ITU-ML5G-PS-010

Forecasting Model for Service Allocation Network Using Traffic Recognition Progress Report

Prepared by

Ghazal Rahmanian, Mohammad Yekta,

Hadi Shahriar Shahhoseini, AmirHossein Jafari

Iran University of Science and Technology Tehran, Iran

ghazal_rahmanian@elec.iust.ac.ir
mohammad_yekta@elec.iust.ac.ir
shahhoseini@iust.ac.ir
amirjafari@iust.ac.ir

September 16, 2021

Executive Summary

This report is prepared to outline our contribution for the challenge of designing a forecasting model for service allocation network using traffic recognition. This challenge has introduced two problem statement with a specific training dataset.

The Project

Based on the description mentioned in the challenge, focusing on the intelligent application demand of networking management and computing resource management, the artificial intelligence technologies, machine learning for instance, provide us with a possible automotive approach for the existing challenges. One of the biggest challenges in this realm is services traffic allocation. Here two problem statements have been proposed regarding the services allocation and traffic forecasting based on user needs for services.
These problem statements are as follows:

1. Proposal with ML model for recognizing the user demands based on the traffic services allocation.
2. Proposal with ML model for traffic forecasting, taking into account traffic types and user demands (in order to future service migration).

The training dataset is a 20GB zipped pcap format data including several different captures.

Our Contribution

The Research procedure consists of three main phases: Data gathering phase, data Preprocessing phase and implementing Machine learning models. The first phase was done by the challenge's creators and they had provided us with a raw dataset. In the second phase, the main target is to derive a matrix of network features in different types of sessions. This matrix is the input to our next phase, implementing ml models for network traffic prediction. The research's main target was to apply a ML model to predict network traffic. In this section we are going to elaborate more on these two phases.

1. Data preprocessing

Data preprocessing is an essential step in data-driven approaches like here; this is because data is often unsuitable for training software algorithms such as machine learning. Network traffic is normally captured in raw PCAP (Packet CAPture) format which is not processable. Hence, it is necessary to prepare the data and transform it into a usable format. We should analyze the contents of PCAP files and generates several attributes (features) such as Source Port, Destination Port, Protocol and etc. The data preprocess phase are as follows:

1.1. Conversion and labeling

The very first step for data preprocessing is converting the pcap file to CSV, making it lighter in volume and usable in our code, making sure the data is in good shape for analysis. One of the main challenges in this research was its high-volume dataset.

The next step after altering the raw dataset, is labeling the data as source IP, destination IP, protocol, source port, destination port and etc.

1.2. Extraction

In this step, the main goal is extracting the required information in each packet to assemble a comprehensive view of packet flow and requests in our network. For this purpose, information of each packet is analyzed to recognize which stage of communication are we in (SYN, ACK, , FIN) or we have a complete or a failed and unanswered request. This is done by inspecting the flags in each request individually and all of them consequently. Here, the outcome is a table of the number of completed connections between different servers and clients in different periods of time. This is done by observing the SYN, ACK and FIN flags in the TCP connections

1.3. Traffic Features Matrix

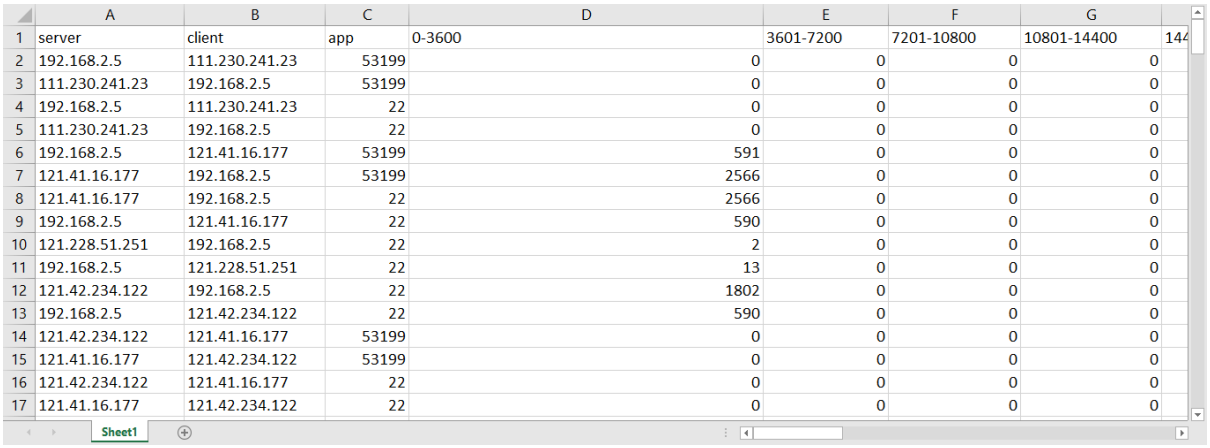
By observing the sequence of requests and their flags and seq and ack number and their window size, we can get to a thorough vision of what exactly is happening in the network and what should we expect in the incoming days and hours.

Once having a comprehensive list of successful or failed and unanswered requests with the help of the extracted data in the previous part, we get to recognize other helpful information regarding the timing of requests.

The timing of request can be observed from different perspectives; the time duration for each successful request to be complete, the number of completed and failed sessions in a period of time, the number of requests in a period time, the volume of data transferred in a specific time or a period of time. This data can also be observed for a specific source and destination IP or for a specific source and destination port.

In this step, we are extracting a matrix from our dataset. In this matrix, rows are defined with three different feature, Server (Destination IP), Client (Source IP) and Application (Destination port). The columns in the matrix are equal period of times.

For completing the matrix, we have taken into account the complete TCP connections (Connections with SYN, ACK and FIN). As it can be seen in the picture, we convert each connection into two consecutive rows in this matrix. In each period of time, we have calculated the difference between the [Ack] number and the [Seq] number between the starting time and the ending time, calculating the volume of transferred data from client to server and vice versa. The first row of each connection includes the difference of acknowledge number and the next row includes the difference of sequence number in a specific period of time for a specific server, client and application. For the second line, we assume our initial Server as client and out previous client as server. Depending on what we want to predict, we can extract the desirable information from this matrix.



	A	B	C	D	E	F	G	
1	server	client	app	0-3600	3601-7200	7201-10800	10801-14400	144
2	192.168.2.5	111.230.241.23	53199	0	0	0	0	
3	111.230.241.23	192.168.2.5	53199	0	0	0	0	
4	192.168.2.5	111.230.241.23	22	0	0	0	0	
5	111.230.241.23	192.168.2.5	22	0	0	0	0	
6	192.168.2.5	121.41.16.177	53199	591	0	0	0	
7	121.41.16.177	192.168.2.5	53199	2566	0	0	0	
8	121.41.16.177	192.168.2.5	22	2566	0	0	0	
9	192.168.2.5	121.41.16.177	22	590	0	0	0	
10	121.228.51.251	192.168.2.5	22	2	0	0	0	
11	192.168.2.5	121.228.51.251	22	13	0	0	0	
12	121.42.234.122	192.168.2.5	22	1802	0	0	0	
13	192.168.2.5	121.42.234.122	22	590	0	0	0	
14	121.42.234.122	121.41.16.177	53199	0	0	0	0	
15	121.41.16.177	121.42.234.122	53199	0	0	0	0	
16	121.42.234.122	121.41.16.177	22	0	0	0	0	
17	121.41.16.177	121.42.234.122	22	0	0	0	0	

Sheet1

2. Machine Learning algorithms

ML is a subsection of AI and also a significant learning method used for several real-time applications. The ML method provides good predictions of network traffic. Thus, it outperforms complex mathematical models. ML methods can be subdivided into supervised, unsupervised, and reinforcement learning techniques.

- Supervised learning is widely used for the classification problem. It can be divided into classification and regression model for traffic prediction and it can be applied when there are labeled data. Support Vector Machine, Neural Network and Naïve Bayesian are common classification methods and there are Linear Regression, Auto Regressive, Support Vector Regression and Neural Network for regression model.

- Unsupervised learning is widely used for the clustering problem. The cluster methods are trained with non-labeled and produce useful patterns. Common clustering algorithms are K-means, Fuzzy Gaussian, Hidden Markov Model and Neural Network.

- Reinforcement learning is a subdivision of ML which studies systems that are able to learn from data. RL has been utilized to construct the systems that learn to perform the task for the nontrivial sequential decision. In reinforcement learning, the agent learns how to execute actions in the environment. Q-Learning, Deep Q Network, SARSA and Monte Carlo are common RL methods.

2.1. ML algorithms for network traffic prediction

Regarding the second problem statement, traffic forecasting, there are two different types of prediction model in terms of time granularity. These models are used for short or long period prediction. In short period prediction model, the time granularity of the forecasting process is in terms of minutes and hours, while in long period prediction model, the time granularity is in terms of days, weeks and months.

The studies show that linear models perform well for short period prediction whereas non-linear models showed better results for long term prediction. However, there are some methods that are combination of linear and non-linear methods. Time series prediction can be used to forecast network traffic patterns at different time intervals. Network traffic can be divided into periods of one minute or one hour. We can also have daily traffic and analyze the traffic of a whole day.

To forecast the next hour traffic of a network, the following are chosen: the traffic of the current hour, the traffic of the same hour in the previous day, the number of sessions, sessions' duration and their volume. The mentioned attributes can be extracted from the dataset. We have chosen supervised learning according to our labeled dataset and in this category, regression method has been chosen to predict the network traffic. From different regression algorithms, Support vector regression (SVR) and Nonlinear Autoregressive neural network (NAR) are applied to forecast the network traffic.

2.2. Support Vector Regression (SVR)

Support vector regression is a type of Support vector machine (SVM) which can be used for time-series forecasting problems. Network traffic prediction is a time series analysis problem, making SVR a suitable solution for this problem. A short-term network traffic prediction model based on SVR combining with particle swarm optimization (PSO) for optimizing SVR parameters (C, γ and ε) is implemented to improve network traffic prediction model.

2.3. Nonlinear Auto Regressive neural network (NAR)

Dynamic neural networks are good at time-series prediction. Nonlinear Auto Regressive neural network (NAR) is one of the algorithms used for time series prediction. In this type

of time series problem, there is only one series involved. The future values of a time series $y(t)$ are predicted only from past values of that series. This form of prediction can be written as follows:

$$y(t) = f(y(t-1), \ldots, y(t-d))$$

3. Simulation and codes

Predicting traffic can be done in two different ways:

1. Predicting the traffic for a specific application in future.

2. Predicting the traffic for a specific server.

Based on the extracted data from matrix, we can calculate the volume of transferred data in our network and predict the traffic.

3.1. Preprocess code

This code gets a csv file as input and gives us a two txt files as output:

- Output1: the series of numbers which indicates the volume of the data transferred in duration of every minute based on the value of ack number
- Output2: the series of numbers which indicates the volume of the data transferred in duration of every minute based on the value of seq number

Csv file is the matrix that its rows are adjusted for different servers, different clients and different applications. There are two different options to choose:

Option a: calculates traffic for a specific application

Option b: calculates traffic for a specific server

We choose the desired option by changing its related value to 1. This code sums up the related rows and gives the result. The outputs will later be used as an input for the SVR and NAR code.

3.2. SVR & PSO code

This code is an implementation of SVR code in python and is combined with PSO. The PSO algorithm optimizes the parameters of SVO which are C, γ and ε and gives an optimum regression equation. The parameters of PSO are set as follows:

Number of particles =120

Number of iterations =100

Number of dimensions =3

Inertia =1

Train data= 70%

Test data =15%

Validation data =15%

This code gets a txt file (output1.txt or output2.txt which are the normalized traffic data) as input and gives the prediction graph and calculates the MSE. The following figure is an example of the mentioned algorithms for one part of the dataset.

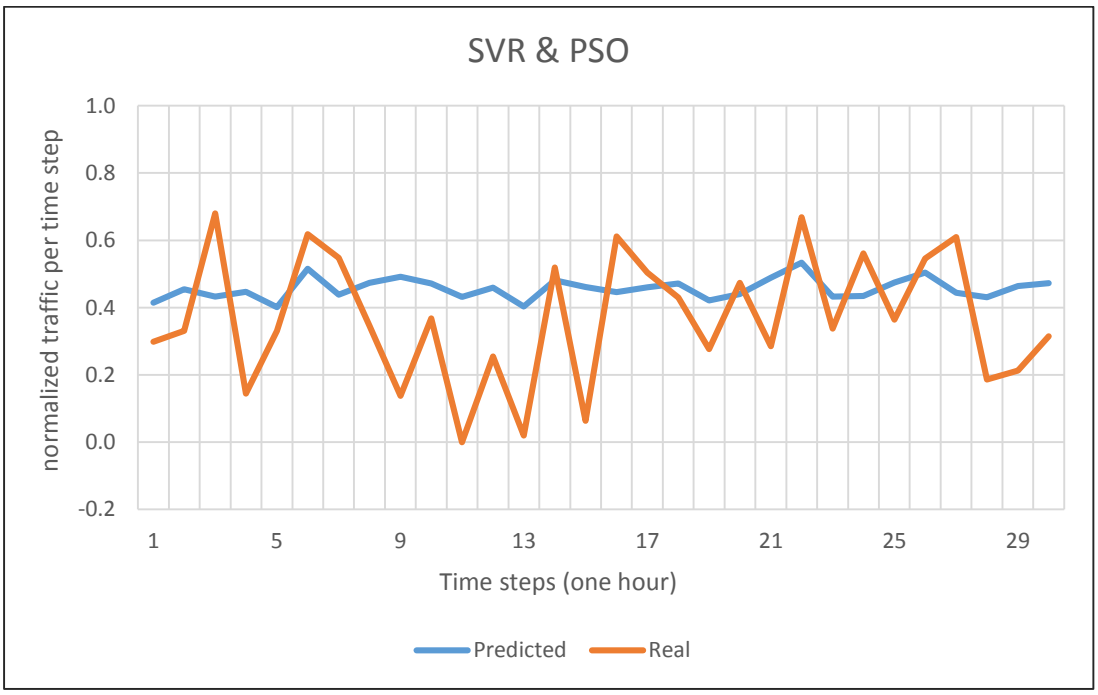


Figure 1 : normalized traffic per time step for SVR & PSO

3.3. NAR code

This code is a combination of auto regressive and neural network which is implemented in MATLAB environment. This code gets a txt file (output_n.txt which is the normalized traffic data) as input and gives us the prediction graph and calculates the MSE. The parameters of PSO are set as follows:

Number of hidden layers =10

Number of iterations =100

Number of delays =24

Train data= 70%

Test data =15%

Validation data =15%

The following figure is an example of the mentioned algorithms for one part of dataset.

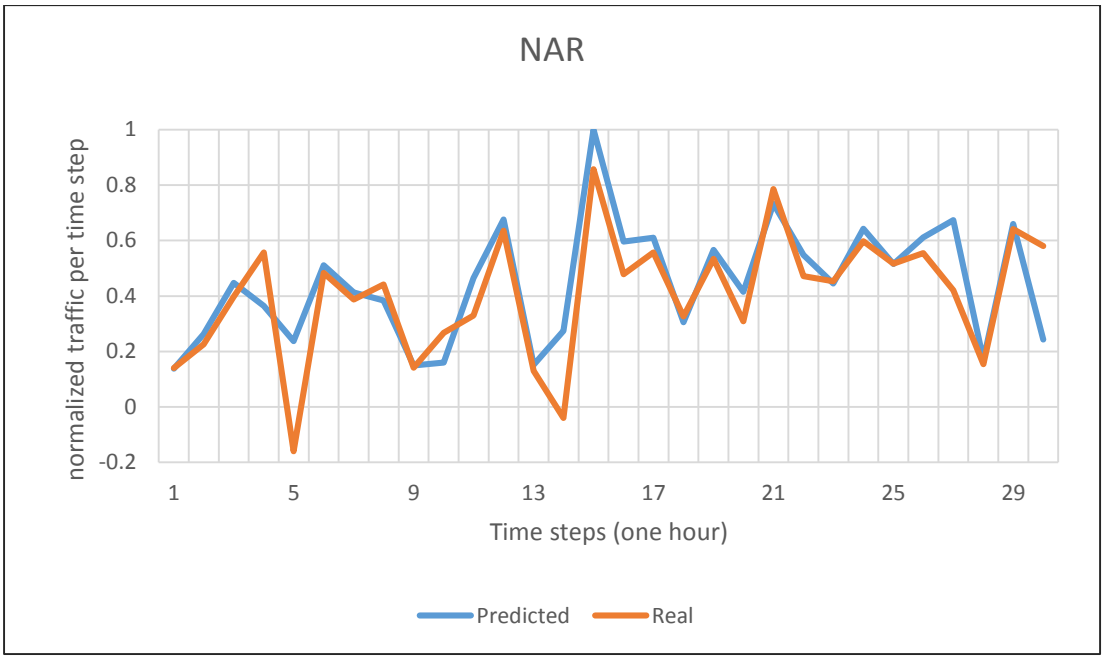


Figure 2 : normalized traffic per time step for NAR

4. Result and evaluation

The preprocessed dataset which is a kind of time series data and contains the traffic of every hour is used as an input for our two ML models (SVR & PSO and NAR). Comparing the figures 1 and 2, we can see that in NAR algorithm, the predicted traffic is closer to the real traffic than in SVR algorithm.

For these two algorithms we have also calculated the Mean Squared Error (MSE). The following are the results:

For SVR & PSO: MSE= 0.0408

For NAR: MSE= 0.0192

Lower MSE in NAR shows that it is the better ML model to predict the network traffic and it can provide us with better results for different use cases.