ISSN NO: 2770-4491
Date of Publication:20-05-2022

Methods for network monitoring to detect anomalies in time series

Alaa Abdulhussein Daleh Al-magsoosi
ITMO University And College for pure science University of Wasit, Iraq
adleah@uowasit.edu.iq

Abstract : The open source RRDtool software solves the challenge of collecting and storing time series data from service networks. However, even an experienced network technician will find it difficult to monitor all relevant service network time series at the same time. The approach is to incorporate a mathematical model into the monitoring application that can automatically detect deviations in the time series. And when we talk about the growing number of models, the choice is difficult, because the chosen model must be compatible with real-time monitoring, in this paper we adopt a new approach which is to integrate the theory of exponential smoothing and intelligent properties in Holt Winters algorithm by using RRDtool as a tool for data collection and representation With real-time schemes, although this technique is not perfect, it is adaptable, effective and successful as a tool for automatic identification of deviant behavior after being trained to identify anomalies through prediction based on past and future steps.

Keywords: RRDtool, Exponential Smoothing, Holt-Winters Forecasting Algorithm, timestamps, One-hot Encoding

1. Introduction

The problem of real-time network service infrastructure control remains a concern for system administrators at the IAP / ISP level. Initially, huge amounts of data were generated based on continuous monitoring of one minute per cycle. To handle a subscriber base of over millions of people, the types of data collected are quite diverse. The Message Production Service tracks SNMP counters across network connections, host data including CPU load and I/O activities, and application object event logs. Every variable is tracked, whether it's the byte flow on the switch port or the computer's CPU load. A time series is generated when requests are handled by a cookie daemon. These time series are an indication of the accuracy of the recorded results of the service [1].

As a result, the first problem is the mechanism of collecting this data and how to store it in a form in which it is easy to find, and after this obstacle is overcome, the network technician can display the graph sequentially based on the real-time monitoring.

Deviant behavior, or differences in the behavior of short-term time-series with unprecedented historical correlations, It is possible that the network technician will disturb the way that the time is arranged according to special classifications that are not important depending on the way the service is monitored in a dynamic way. Performance throttling, application component failure, or system failure can be indicators of abnormal activity to expect abnormal behavior in some situations but not in others.

The second problem of network monitoring is detecting skewed behavior in hundreds of service network time series automatically. Once this activity is observed, the system can send an alert to the technician indicating the potential problem and this may provide some capability through existing software tools, however these solutions are mainly based on basic rules or thresholds. The current rules are considered a suitable solution for many applications, but they are unable to identify more variables in the behavior of anomalous data and are unable to infer new criteria instead of the ones that are routinely followed [2, 3].

2. Related Work

Network traffic has been extensively studied for several decades. Our work comes as a supplement, by providing a pivotal study of different web applications. The focus of applications on a single protocol stack has been observed and predicted for a while. Tunnels is a way to bypass the restrictive policies of rewalls by

ISSN NO: 2770-4491
Date of Publication:20-05-2022

encapsulating a blocked application in HTTP requests and responses. HTTP Tunneling [4, 5] is a way to bypass the restrictive policies of rewalls by encapsulating a blocked application in HTTP requests and responses.

3. Identifying and Defining Abnormal Behavior

If we assume that there is a statistical model that describes the behavior of a time series where the characteristics of this model cover the requirements that we need, it is possible with such a model to define abnormal behavior as behavior that is incompatible with the model and may be misrepresented [6].
Of course, abnormal behavior concerning The statistical model may be wrong to produce an effective solution occurrence that the technician is interested in. If it doesn't, it's considered a false positive. The ideal situation is to have as few false positives as possible while yet recognizing all occurrences of genuine relevance. This ideal, however, is rarely realized. There is a trade-off in most detection methods between selectivity and minimizing false positives.
It is sometimes known as accuracy, specificity, and ability to detect anomalies sensitively and positively. It can also be referred to as the art of recall. It is necessary for the statistical model to be familiar with these difficulties. If the statistical model is viewed as abnormal behavior, it becomes a screening tool and not a substitute for the network technician.
Many of the changing time series in the service network display the following characteristics, which the model should take into account as The time component, which means that the rise in requests may be gradual or rapid during a 24-hour period, which may be attributable to an increase in the number of subscribers or unusual activity inside the network (eg, bytes per second increases each day in the morning hours, peaks in the afternoon and decreases late at night) [6].
In addition to the above, the design of the model must focus on the context of real-time monitoring, and it can be difficult for network technicians to understand these models, which can be complex computations compared to the real-time implementation.

3.1. Model Summary

We can divide the process of detecting abnormal behavior into three basic stages, and they can be summarized as follows:
• Using an algorithm to predict future time series values.
• Measure the extent of the deviation and compare the values that can be expected with what is actually observed.
• Establish a mechanism to determine the observed value and to verify the sequence of values and classify them in terms of deviation from the expected values.
In this model, we use the Holt-Winters algorithm to predict outliers, which has the advantage of working with models increasingly on the exponential smoothing theory. n1… nt-1,nt, nt + 1 ... represent the sequence of time values obtained from follow-up at some specific time period and compiled by the RRD tool Zalti of its features identifies irregular time series within a regular time period and schedule[7]. Then we use programming languages to train the algorithm to indicate the period of the seasonal trend and discover the behavior of the data in advance steps.

3.2.Exponential Homogeneity Theory

Exponential smoothing theory is a straightforward approach to predicting the next value in time series of present value and predictable values., Prediction is defined as the average and weighted average of the data in the time series of past outcomes [8, 9]. The basis for understanding exponential homogeneity depends on the fact that the present value is the way to predict the future value.
Since the prediction in the next steps is derived by studying the current prediction with the obtained value, on this conclusion it is a kind of incremental algorithm.

ISSN NO: 2770-4491
Date of Publication:20-05-2022

3.3.Holt-Winters

Forecasting using the Holt-Winters algorithm is a complex method for its dependence on the theory of exponential smoothing, where the work of Holt-Winters is summarized by the dependence on the achieved time series and thus we can summarize it with three components: the linear basis, the trend of the line, and the seasonal effects. The change of components is a matter of time, and it occurs through the gradual updating of the components using the theory of exponential smoothing[10].

The forecast is made up of three parts:

$$y^{f+1} = a_f + b_f + c_{f+1-m}$$

The following are the update formulae for the three components, or coefficients a, b, and c:

• Starting point:

$$a_f = a\left(y_f \ - c_{f-m}\right) + (1-a)\left(a_{f-1} + b_{f-1}\right)$$

` • Slope:

$$b_f = \beta\left(a_f - a_{f-1}\right) + (1-\beta)b_{f-1}$$

• Seasonal Pattern:

$$c_f = y\left(y_f - a_f\right) + (1-y)c_{f-m}$$

Auto-update coefficients as in exponential normalization, representing the current mean of the prediction which results from the observed value yt through the model parameter (α, β, γ). Remember that m represents the period of the seasonal cycle, and thus represents the seasonal coefficient at for time t until it is the last coefficient extracted for the same time point in the seasonal cycle [11, 12].

The observed value is modified with the available mechanisms of the seasonal coefficient to reach the best estimate of the new base value (ct-m), and the sloping value is added to the baseline coefficient and the change must be taken into account in the linear trend level. The result for the difference between the new and old baseline is the value of the slope updater. The new estimate of the seasonal component can be considered as a tool of the difference between the observed values and the baseline value [13].

Holt-Winters Forecasting may also anticipate a time series longer in the future than a single time step. This multi-step prediction method includes a technique for dealing with missing data.

3.4.Detection of Abnormal Behavior

Checking the results of the observed values of the time series is a kind of bypassing the confidence zone, a straightforward approach to anomaly detection, However, this process frequently produces a large number of false positives. A moving window with a set number of observations is a more resilient technique. If the number of infractions surpasses a certain level[14], an alert for unusual conduct is triggered. Formally, a violation is defined as an observation yt that falls beyond the interval:

$$y\hat{}_f - \delta . d_{f-m}, y^{\hat{}}{}_f + \delta . d_{f-m}$$

A failure is defined as having more than a given number of threshold violations within a timeframe of a certain number of observations.

3.5.Seasonal Cycle and Variation Smoothing on a Temporal Scale

Calculations must be made for all components of the vector, seasonal coefficients, separately, as previously mentioned. It is acceptable to infer that the seasonal influence is a smooth function across time, rather than a sequence of discontinuous points[15, 16]. A similar argument may be made for seasonal variances. It is

ISSN NO: 2770-4491
Date of Publication:20-05-2022

important to note that despite the smoothness of the exponential smoothing theory in managing seasonal cycles, it fails to make temporal smoothing logical during the operation of seasonal cycles..

3.6.Model Parameter Selection

For the model to perform properly, the model parameters must be established and tweaked. Even when limited to data for a single variable, there is no one best set of values. This is due to the model's interaction of many factors[17, 18].

We have to keep in mind some notes about the sequence: nt and nt + 1. While updating the exponential smoothing, (intersection coefficients), (slope) and (seasonality) investigate some of the differences between yt and nt+1. Since it is acceptable to assume that the differences are some kind of noise, the changes made to the coefficients do not require all of these calculations. And the differences between the values of yt and nt + 1. The determination of the rate depends on the percentage of differences assigned to the measurement of a baseline and linear trend variables and a variable seasonal coefficient.

4. How to create a round-robin database

The purpose of creating an RRD is to detect anomalies and to make the task of an inexperienced user easier, the build command RRDtool supports the implicit creation of HWPREDICT, SEASONAL, DEVPREDICT, DEVSEASONAL and FAILURES RRAs when HWPREDICT is specified alone[19], in addition to explicit creation of the HWPREDICT, SEASONAL, DEVPREDICT, DEVSEASONAL, and FAILURES RRAs.

4.1.more clarification:

• The rows represent the number of instances of a forecast that will be stored and where the resulting number is greater than the seasonal period. The number will be within a number of DEVPREDICT RRA rows.

• The intersection adjustment parameter must be between 0 and 1. for the same value to be applied..

• The value of beta represents the adjustment parameter of the slope, and it must be between 0 and 1.

• The number of points in the raw data for the seasonal period can be called the term period. This is the number of RRA rows for the SEASONAL and DEVSEASONAL RRAs.

The following is the script used to create the database for the model:

```
#! /bin/bash
rrdtool create database.rrd --step 60 --start N DS: traffic_in:COUNTER:120:0:U \
DS:traffic_out:COUNTER:120:0:U RRA:AVERAGE:0.5:1:1440 RRA:HWPREDICT:1440:0.1:0.0035:1
```

Listing 1: Script for Database Creation

These tags accept an individual entry in a way that makes it compatible with the previous parameters in the model selection section, The adaptation parameters of both SEASONAL and DEVSEASONAL RRAs will be renamed when the gamma flag is set. Deltapos To refactor them, we denote the scale parameters of the two domains, assuming their default value is 2 ..

4.2.Daily observation representation

We now have a new layout option. It creates hashes for non-zero values in the DEF and CDEF parameters. The third option specifies the length of the sign in terms of percentage and decimal places for the y axis. Since the inaccuracies are equal to 1.0 and 100% in terms of the length of the y-axis, the parameters draw a line perpendicular to the histogram [20, 21].

ISSN NO: 2770-4491
Date of Publication:20-05-2022

4.3.Initialization

Execution, as mentioned in the Configuration sections, Its purpose is to use initialization and bootstrap. Where the intercept coefficient is set to the first value so that the slope is set to the zero value based on our assumption that the linear trend as time passes becomes close to zero , And the time it takes for the algorithm to make the values deviate from zero will be calculated with the help of the seasonal adjustment factor, gamma. The parameters are initialized during the sum of the seasonal cycle of the resulting values. And then the seasonal variations of the following seasonal cycle of data can be observed, [22, 23]. After that, the implementation of the necessary initialization of the transactions begins as possible.

4.4.Script for Assigning A value to Databases

The bash script used to populate the database with an update interval of 60 seconds would be shown below; values and parameters can be changed as needed.

```
#! /bin/bash

While true;do
traffic_in=‘snmpwalk -OQEav -v2c -c <community> <ipaddress> [:<dest_port>] [oid] ‘
traffic_out=‘snmpwalk -OQEav -v2c -c <community> <ipaddress> [:<dest_port>] [oid] ‘
rrdtool update database.rrd N: $traffic_in:$traffic_out
sleep 60
done
```

Listing 2: Script to populate databases with values

4.5.Graph Displaying With HWPREDICT Implemented

The python script that updates the rrd database and draws the graphs in 1 hour, 2 hour, and 24 hour intervals, polling data from the database [24], is shown below. The code is well commented to make it easy to follow.

```
  while True:
      enddate = int(time.mktime(time.localtime()))
      os.system( "traffic_in=`snmpwalk -OQEav -v2c -c Test 10.10.10.1
1.3.6.1.2.1.31.1.1.1.6.1`;traffic_out=`snmpwalk - OQEav -v2c -c Test 10.10.10.1
1.3.6.1.2.1.31.1.1.1.10.1`;rrdtool update {} N:$traffic_in:$traffic_out".format( rrdpath))
      gen_image(rrdpath, pngpath, 'data_1h', width, height, 3600, enddate, 'now-1hour')
      gen_image(rrdpath, pngpath, 'data_2h', width, height, 7200, enddate, 'now-2hour')
      gen_image(rrdpath, pngpath, 'data_1d', width, height, 86400, enddate, 'now-24hour')
      time.sleep(60)
```

Listing 3: Script to draws the graphs

4.6. Activate the code

To run the code in the background in order for the graph to continue without stopping while monitoring the data, the following code must be used through the Linux system [25].

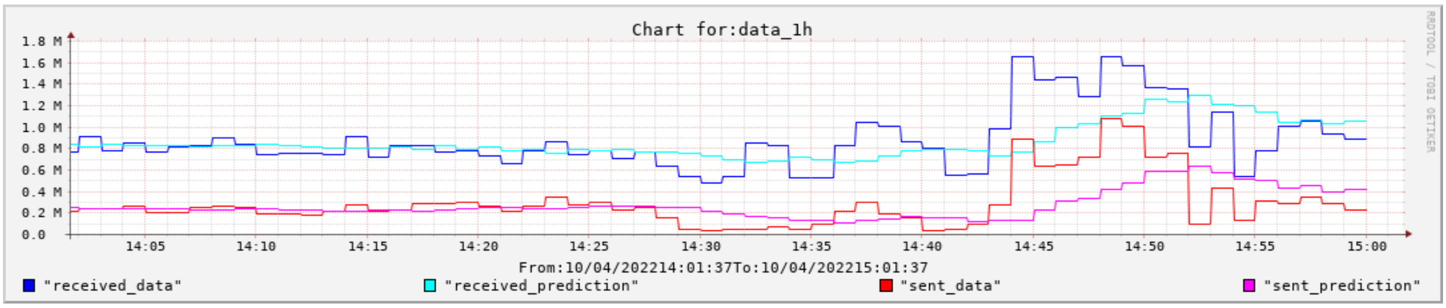
screen -dmSL workstuff python3 *file_name*.py / (root link for rrd file surse) /*rrd_file_name*.rrd /(root link for saved graph)/

Listing 4: Script to run order for the graph to continue without stopping in background

ISSN NO: 2770-4491
Date of Publication:20-05-2022

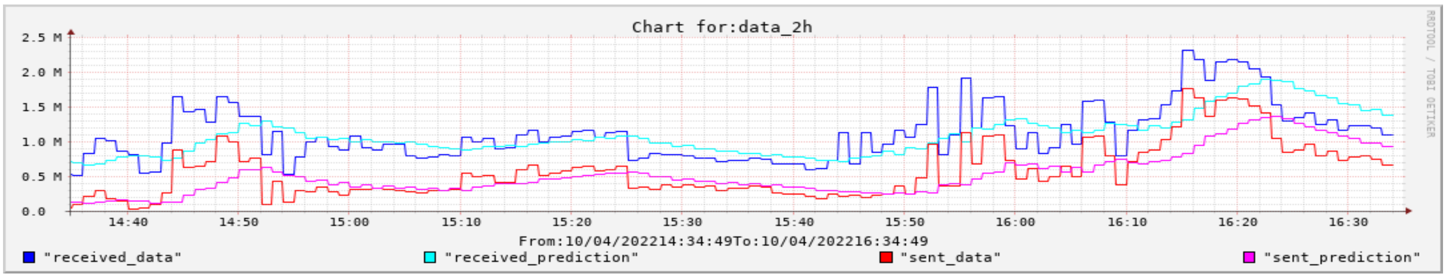
4.7.View results

The graphical results representing the real time of the monitoring are displayed through the creation of the website, where the website displays the images in the following form.



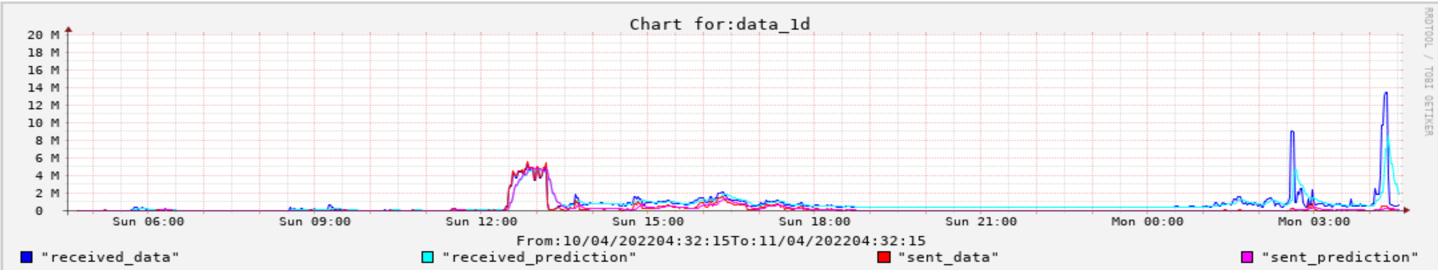
Figr 1: representing the real time of the monitoring for 1 hour

Where the Figr 1 represents the monitoring of data traffic based on the SNMP protocol for a period of one hour.



Figr 2: representing the real time of the monitoring for two hours

As shown in Figr 2, the monitoring of data traffic based on the snmp protocol for a period of two hours.



Figr 3: representing the real time of the monitoring for 24 hours

After completing the monitoring for 24 continuous hours, as shown in Figr 3, the result was a dataset that was collected using the RRDtool tool with the help of snmp protocol.

5. Extraction of RRD Database Data into CSV for Machine Learning

To get the data into CSV format, first convert the rrd database to XML using the command:

`rrdtool dump "rrd database name"` on the terminal.

After that, we were able to obtain a classifier for the data as follows as shown in Figr 4.

 ISSN NO: 2770-4491
Date of Publication:20-05-2022

date	taime	zone_uts	epoch	Traffic_in	Traffic_out
1/2/2022	9:12:00	EST	1641132720	2.37E+04	1.01E+04
1/2/2022	9:13:00	EST	1641132780	2.62E+04	1.04E+04
1/2/2022	9:14:00	EST	1641132840	2.64E+04	8.41E+03
1/2/2022	9:15:00	EST	1641132900	2.24E+04	8.38E+03
1/2/2022	9:16:00	EST	1641132960	2.27E+04	8.65E+03
1/2/2022	9:17:00	EST	1641133020	2.54E+04	9.02E+03
1/2/2022	9:18:00	EST	1641133080	2.72E+04	8.71E+03
1/2/2022	9:19:00	EST	1641133140	2.28E+04	8.46E+03
1/2/2022	9:20:00	EST	1641133200	2.25E+04	8.49E+03
1/2/2022	9:21:00	EST	1641133260	2.46E+04	8.47E+03
1/2/2022	9:22:00	EST	1641133320	2.72E+04	8.76E+03
1/2/2022	9:23:00	EST	1641133380	2.29E+04	8.77E+03
1/2/2022	9:24:00	EST	1641133440	2.28E+04	8.67E+03
1/2/2022	9:25:00	EST	1641133500	2.24E+04	8.29E+03

Figr 4: classifier for the data

5.1.Overdue method as features

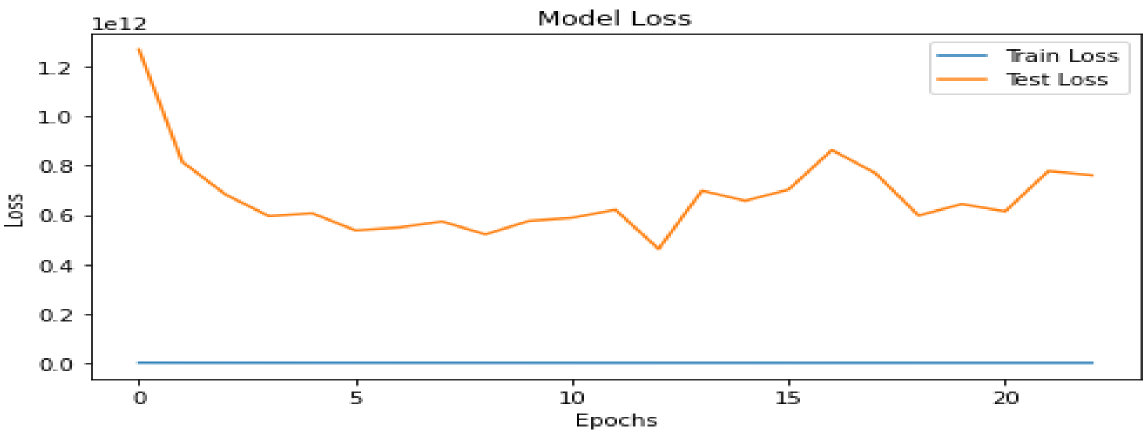
Using time steps as a feature, the prediction is based on the value X (t + n), from which we conclude that n represents t, X + 1, ... and X (t + n-1). Then the goal becomes to create a set of columns based on the previous observations. [26], Using the Pandas library and freeing from the results by shift change all values to columns, and to use for loops to create a series of predictions based on previous observations, we shift the values in column n and remove the n columns with a delay in starting and relying on a new point.

Relying on this, set the number of features and forecasts to determine the new expected value, in other words the delayed observations can be up to 100.

5.2.Features arising from the use of timestamps

feature engineering is more of an art than a science [27]. however some general rules can guide data scientists and the like, My goal in this section is not to delve into all of these practices but to show a couple of them, in fact feature engineering is highly dependent on the area to be delved into, which may require creating a different set of features for the task at hand.

Using a set of data concatenated in univariate time units, it makes sense to create date and time features, and since we've already converted the values into an index from the date and time, which results in a series of objects, we can easily create new features from the index values, such as hours of days[28], as shown in Figr 5.

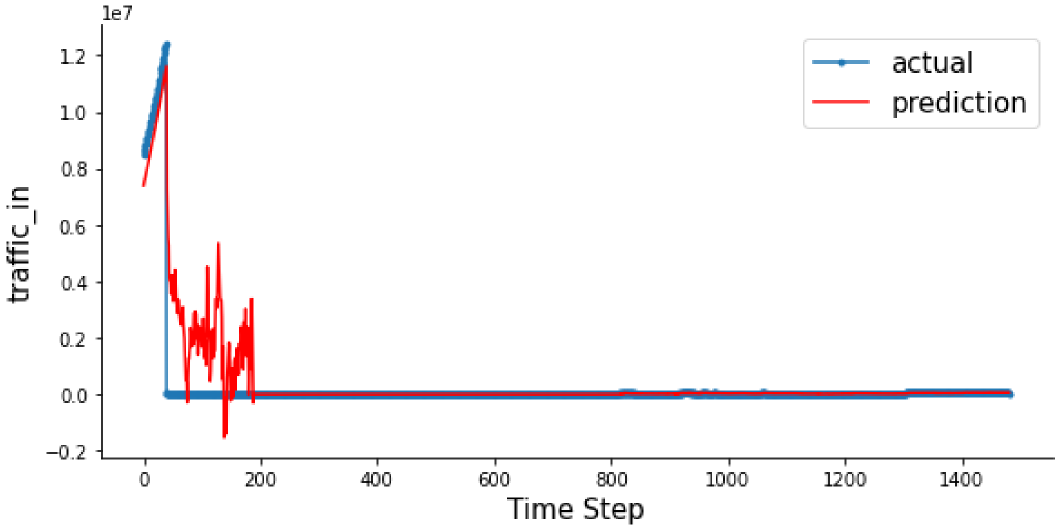


Figr 5: series of DateTime

Although sending date and time characteristics to the model, it will be difficult for the model to understand their interrelationships, it is very easy to notice that the recorded data follow relatively cyclical patterns, but this may not be obvious, algorithms designed to understand the first month of the year have been developed

ISSN NO: 2770-4491
Date of Publication:20-05-2022

after The twelfth month, this makes robust feature engineering a prerequisite for building deep learning models, especially classic machine learning models.



Figr 6: series of Date Time after training

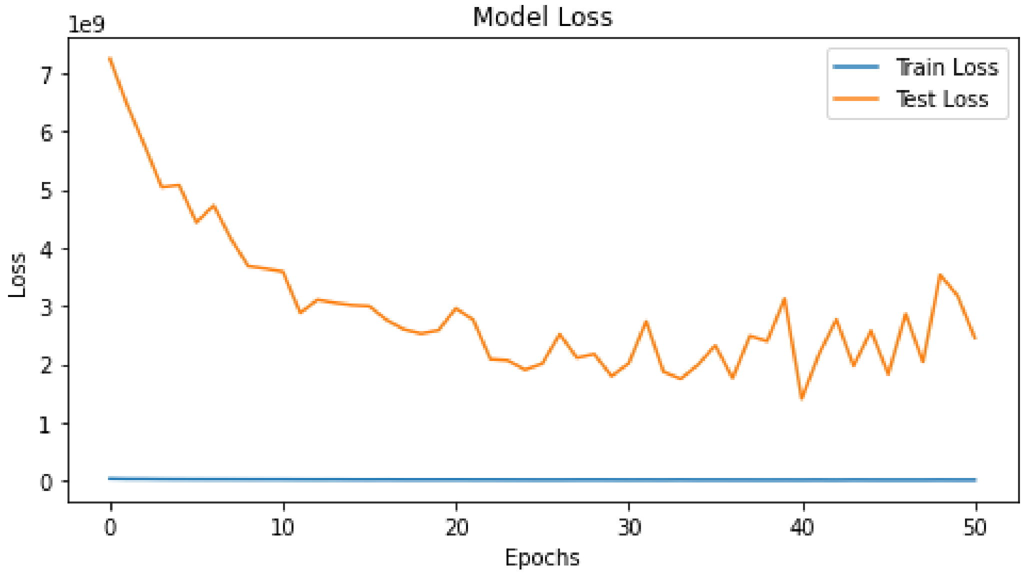
The future thinking would be to consider how to perform a one-time separation of features into two versions in decision tree based models, and since these features will form their splits according to one feature at a time, so they will fail to use the same features twice, these models are usually powerful enough It is enough to deal with such divisions, and with the result the distribution of values will be as shown in Table 1.

Date_Time	Zone-utc	Epoch	Traffic_in	Traffic_out
2022-01-02 09:12:00	EST	1641132720	23700.0	10100.0
2022-01-02 09:13:00	EST	1641132780	26200.0	10400.0
2022-01-02 09:14:00	EST	1641132840	26400.0	8410.0
2022-01-02 09:15:00	EST	1641132900	22400.0	8380.0
2022-01-02 09:16:00	EST	1641132960	22700.0	8650.0

Table 1 : tree-based models

5.3.One-hot Encoding theory

5.4.One way to encode DateTime features is to treat them as categorical variables and add a new binary variable for each unique value, widely known as one hot encoding, let's say you apply one hot encoding to the month column, which ranges from 1 to 12 . In this case create 12 columns for months, ie [January, February, ... December], only one of these columns has the value 1 while the rest are zero[29]. Hence some DateTime values must contain the second of these columns encoded as 1, as in [0, 1, ... 0] Using the get_dummies method from Pandas, we can quickly create columns encoded from a given dataset shown in the figure 7.



Figr 7: Data after using One-hot Encoding theory

6. **Conclusions**.

Modern network management solutions must be able to meet current and future requirements. With resource-efficient and easy-to-use solutions, proactive network management can be achieved, and although many network management tools are available, network administrators need to ensure that the solution they choose ensures that managed networks are protected from potential attacks. Network management security is critical to the security of the entire network, and strong security is required for network management protocols and applications. The results we got were acceptable if we used the second generation of SNMP v2c, which is characterized by ease and medium security.

Future research will focus on network management weaknesses as well as remedies for security threats. Moreover, further studies will be conducted on the latest version of SNMP v3, which potentially provides improved security protection.

References

1. J. R. Langevin, M. T. McCaul, S. Charney, and H. Raduege, "Securing cyberspace for the 44th presidency," CENTER FOR STRATEGIC AND INTERNATIONAL STUDIES WASHINGTON DC2008.
2. R. J. R. Langevin, R. M. T. McCaul, S. Charney, L. G. H. Raduege, and J. A. Lewis, "Cybersecurity two years later," *Washington, DC: Center for Strategic and International Studies*, 2011.
3. R. M. Al_airaji, H. Salim, and A. M. Alaidi, "Automated Cheating Detection based on Video Surveillance in the Examination Classes," *iJIM*, vol. 16, no. 10, 2022.
4. B. Augustin and A. Mellouk, "On traffic patterns of http applications," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, 2011, pp. 1-6: IEEE.
5. H. T. ALRikabi and H. T. Hazim, "Enhanced Data Security of Communication System Using Combined Encryption and Steganography," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, 2021.
6. A. Edition, "Information Operations Primer," 2006.
7. W. Jiang, X. Wu, Y. Gong, W. Yu, and X. Zhong, "Holt–Winters smoothing enhanced by fruit fly optimization algorithm to forecast monthly electricity consumption," *Energy*, vol. 193, p. 116779, 2020.
8. E. M. de Oliveira and F. L. C. Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776-788, 2018.

9. I. A. Aljazaery, H. T. Salim ALRikabi, and A. H. M. Alaidi, "Encryption of Color Image Based on DNA Strand and Exponential Factor," *International Journal of Online Biomedical Engineering,* vol. 18, no. 3, pp. 101-113, 2022.
10. C. Liu, B. Sun, C. Zhang, and F. Li, "A hybrid prediction model for residential electricity consumption using holt-winters and extreme learning machine," *Applied energy,* vol. 275, p. 115383, 2020.
11. Y. Zhu, Y. Zhao, J. Zhang, N. Geng, and D. Huang, "Spring onion seed demand forecasting using a hybrid Holt-Winters and support vector machine model," *PloS one,* vol. 14, no. 7, p. e0219889, 2019.
12. I. A. A. Haider TH. Salim ALRikabi, Jaafar Sadiq Qateef, Abdul Hadi M. Alaidi, Roa'a M. Al_airaji, "Face Patterns Analysis and recognition System based on Quantum Neural Network QNN," *International Journal of Interactive Mobile Technologies (iJIM),* vol. 16, no. 9, 2022.
13. D. Bissing, M. T. Klein, R. A. Chinnathambi, D. F. Selvaraj, and P. Ranganathan, "A hybrid regression model for day-ahead energy price forecasting," *Ieee Access,* vol. 7, pp. 36833-36842, 2019.
14. J. Zhang, C. Wu, Y. Wang, and P. Wang, "Detection of abnormal behavior in narrow scene with perspective distortion," *Machine Vision and Applications,* vol. 30, no. 5, pp. 987-998, 2019.
15. P. K. Dunstan, S. D. Foster, E. King, J. Risbey, T. J. O’Kane, D. Monselesan, A. J. Hobday, J. R. Hartog, and P. A. Thompson, "Global patterns of change and variation in sea surface temperature and chlorophyll a," *Scientific reports,* vol. 8, no. 1, pp. 1-9, 2018.
16. A. F. Al-zubidi, N. F. AL-Bakri, R. K. Hasoun, and S. H. Hashim, "Mobile Application to Detect Covid-19 Pandemic by Using Classification Techniques: Proposed System," *International Journal of Interactive Mobile Technologies,* vol. 15, no. 16, 2021.
17. C. Dupke, C. Bonenfant, B. Reineking, R. Hable, T. Zeppenfeld, M. Ewald, and M. Heurich, "Habitat selection by a large herbivore at multiple spatial and temporal scales is primarily governed by food resources," *Ecography,* vol. 40, no. 8, pp. 1014-1027, 2017.
18. S. H. Abbood, H. N. Abdull Hamed, M. S. Mohd Rahim, A. H. M. Alaidi, and H. T. Salim ALRikabi, "DR-LL Gan: Diabetic Retinopathy Lesions Synthesis using Generative Adversarial Network," *International Journal of Online and Biomedical Engineering(iJOE),* vol. 18, no. 3, 2022.
19. A. Sahai, "VM Power Prediction in Distributed Systems for Maximizing Renewable Energy Usage," *arXiv preprint arXiv:1402.5642,* 2014.
20. J. D. Brutlag, "Aberrant behavior detection in time series for network service monitoring," in *14th Systems Administration Conference (LISA 2000)*, 2000.
21. N. Alseelawi, and H. T. Hazim, "A Novel Method of Multimodal Medical Image Fusion Based on Hybrid Approach of NSCT and DTCWT," *International Journal of Online Biomedical Engineering,* vol. 18, no. 3, 2022.
22. T. Öhman, L. Gawriyski, S. Miettinen, M. Varjosalo, and S. Loukovaara, "Molecular pathogenesis of rhegmatogenous retinal detachment," *Scientific reports,* vol. 11, no. 1, pp. 1-15, 2021.
23. M. K. Abdul-Hussein and H. Alrikabi, "Evaluation of the Interference’s Impact of Cooperative Surveillance Systems Signals Processing for Healthcare," *International Journal of Online and Biomedical Engineering(iJOE),* vol. 18, no. 3, pp. 43-59, 2022.
24. D. Zhang, N. Jayasena, A. Lyashevsky, J. L. Greathouse, L. Xu, and M. Ignatowski, "TOP-PIM: Throughput-oriented programmable processing in memory," in *Proceedings of the 23rd international symposium on High-performance parallel and distributed computing*, 2014, pp. 85-98.
25. S. T. Roberts, "Behind the screen," in *Behind the Screen*: Yale University Press, 2019.
26. S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies,* vol. 11, no. 7, p. 1636, 2018.
27. H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in iot systems," *IEEE Internet of Things Journal,* vol. 8, no. 13, pp. 10327-10335, 2020.
28. X. Tang, T. Qian, and Z. You, "Generating behavior features for cold-start spam review detection with adversarial learning," *Information Sciences,* vol. 526, pp. 274-288, 2020.
29. P. Rodríguez, M. A. Bautista, J. Gonzàlez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing,* vol. 75, pp. 21-31, 2018.