Solving Network Alarm flooding through Syslog analysis

"Too many alarms is just the symptom, not the problem"

Chandrashekar Vasudevan
Advanced Services, Cisco
Bangalore, India
chavasud@cisco.com

Devesh Srivastava
Advanced Services, Cisco
Bangalore, India
devesriv@cisco.com

Suresh Shetty
Manager, Advanced Services, Cisco
Bangalore, India
surshett@cisco.com

Lavanya Gopalakrishnan
Director, Advanced Services, Cisco
Bangalore, India
lgopalak@cisco.com

KEYWORDS: SYSLOG DATA MINING, ASSOCIATION RULES, NETWORK ALARMS

Abstract

In any kind of IP Network, an Alarm is an essential element. An Alarm is an event to which an operator must react, respond and acknowledge. But the excess of these alarms can create havoc and impact the decision-making process. How to control flooding of alarms and identify relevant alarms? This paper illustrates the same through the use of SysLog messages and Associations between them.

Introduction

A widespread issue of alarm overload has been induced due to growth of IP Network services. A Network management system is required to perform proactive operations which can quickly detect the signs of critical failures and avoid future issues. To achieve this goal it is important to receive relevant indications through the alarms generated by network elements (e.g. switches, routers). Each element has to monitor an increasingly wider area and consequently deal with more alarms, most of them could be redundant.

Business Problem

The operators when presented with too many alarms may overlook an important indicator of an abnormal situation and may not take necessary action. Without rigorous alarm streamlining, the flooding of alarms becomes a very serious problem, resulting in the increase in risk of major failures and infrastructure damage eventually leading to operational losses.

Network log data, including router syslog and alert logs generated in Network Management Systems (NMSs) are rich sources for performing analysis on alarms. However, it has become impossible to find genuinely important messages that lead to serious problems due to the large volume and complexity of SysLog data.

SysLog data contains information about the events that a router encounters over time. Each event is recorded with a timestamp and also a description of the event. Syslogs are essentially a time related series of events that occur on various network device. This regular recording of events can be used to determine the cause of a network failure at any point in time by going over the SysLog events up to that point.

ROUTER_NAME	MSG_TYPE	COUNT	SEVERITY
ten03.southgate.mi.michigan.telecom.net	UBR10000-6-LB_MODEM_SUCCESS	42	6
ten03.southgate.mi.michigan.telecom.net	UBR10000-4-BADTXOFFSET 13016	4	
ten03.southgate.mi.michigan.telecom.net	SYS-5-PRIV_I 3 5		
ten03.southgate.mi.michigan.telecom.net	UBR10000-4-DSX_MSG_ERROR	13	4
ten03.southgate.mi.michigan.telecom.net	UBR10000-5-EXPIREDBPITIMEOUT	4	5
ten03.southgate.mi.michigan.telecom.net	PARSER-5-CFGLOG_LOGGEDCMD	1	5
ten03.southgate.mi.michigan.telecom.net	SEC_LOGIN-5-LOGIN_SUCCESS	37	5
ten03.southgate.mi.michigan.telecom.net	UBR10000-4-REGISTRATION_BEFORE_TFTP	1	4

Figure 1: Typical SysLog messages

Proposed Solution

The abnormal behavior of SysLog messages not only depends on keywords in the messages (e.g. FAILURE, ERR) but also on log generation patterns, which form some episodes. These episodes describe temporal relationships between logs (e.g. "IF a certain combination of alarms occurs within a time period, THEN another combination of alarms will occur within a time period").

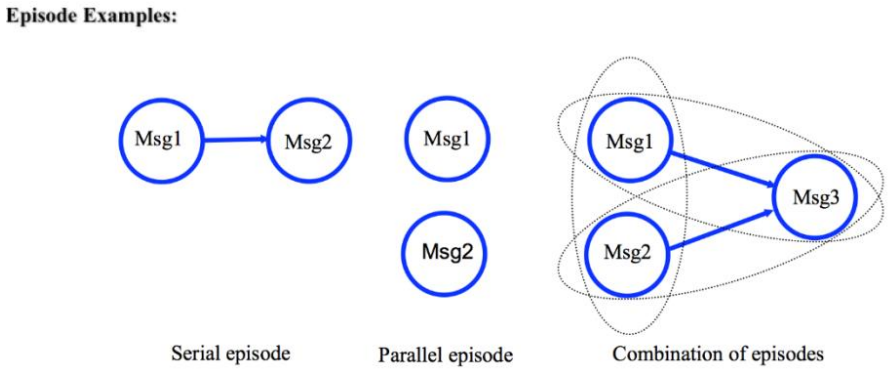


Figure 2: Episode examples

We propose a SysLog analysis mechanism for finding association rules with various messages in the Syslog forming some episodes together at a particular instance. These rules include genuinely important logs with high probability of association in a particular episode. This helps in identifying most relevant alarms along with their occurrence pattern which would guide to find out the root cause.

The uniqueness of this methodology is to automatically learn the relationship between critical failures and log messages irrespective of prior knowledge on logs and generate the appropriate rules. This approach does not require deep knowledge on network domain as well.

Methodology

As per our proposed solution, to generate relationships among a set of SysLog messages to form episodes, there need to be two major grouping:

i. Subnet Grouping: This identifies the devices which need to be grouped together for generating episodes among their Syslog messages. E.g. ***{Msg_A_Device_1, Msg_B_Device_1 => Msg_C_Device_2}.*** This basically gives us a situation where *Msg_A* on *Device_1* and *Msg_B* on *Device_1* has led to *Msg_C* on *Device_2* with a high probability. This allows us to explain the occurrence of *Msg_C* on *Device_2* provided that they are part of the same sub-network and have an influence over one another

ii. Transaction Grouping: Transaction identifier for grouping together a set of messages as belonging to one set. Since SysLog messages are all individual time-stamp based events, there is no way of identifying transactions naturally. Transactions are essential for Association Rule mining

To solve the Device Grouping issue, we decided to depend on the design of the network which naturally consists of sub-networks and devices which are inter-connected to one other. The area based grouping ensures that all devices are part of the same sub-network and there is a high chance of having inter-associations. So when generating Association Rules, we take the SysLogs for all the devices in the identified area together and suffix the message events (*Msg_A*) with the device name (*Device_1*) to arrive at a compound event like ***Msg_A_Device_1***.

To solve the Transaction Grouping issue, we adopt a Sliding Window [1] approach to identify windows of events that we can consider as individual transactions for our purpose. As illustrated below:

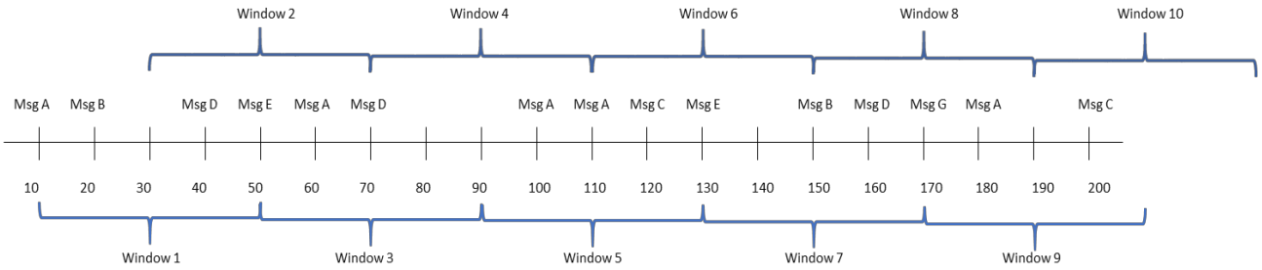


Figure 3: Sliding window process

So as an example, keeping a window size of 40 and a step size of 20, we can identify ten windows in the given time-line and consider these windows as individual transactions for generating the Association Rules. We actually used a window size of 300 seconds and a step size of 150 seconds.

The SysLog messages were stored in Hadoop clusters and retrieved through Hive. The data pre-processing, exploratory analysis and Association Model building was done in R. Specifically, the NB Rule Miner Algorithm [2] was used for Association Modelling.

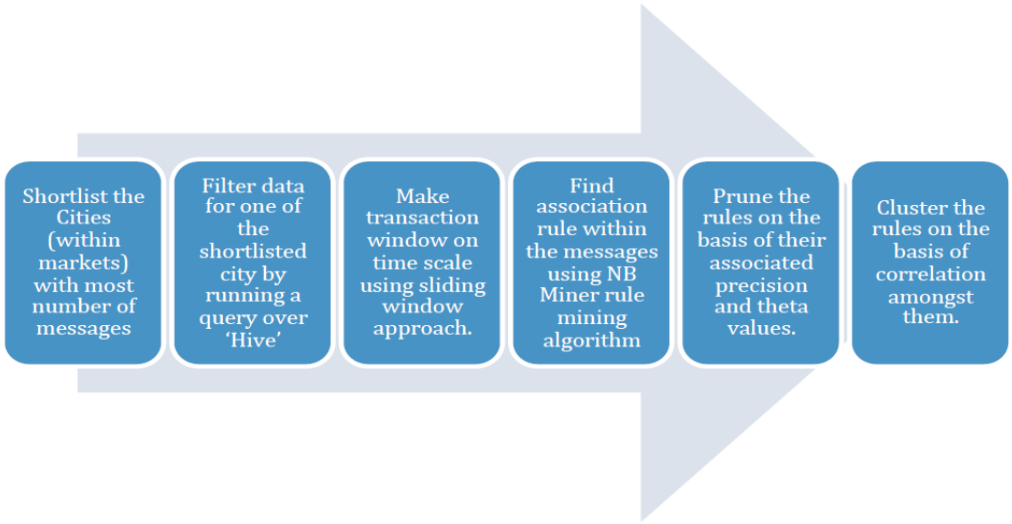


Figure 4: Process Flow of Rule generation

Findings

There can be multiple episodes which can be used to explain certain situations on the network and result in creating Alarms. A couple of instances are explained below:

I. The below generated rules through our process has discovered episodes that can be a potential trigger to generate a Security Alarm

Rule	Antecedent		Consequent
1	ACL-IPV4_ACL-6-IPACCESSLOGP-device1, IP-DHCPD-3-NOPACKET-device1, IP-DHCPD-3-NOPACKET-device2, SECURITY-login-4-AUTHEN_FAILED-device2, SECURITY-SSHD-3-ERR_DETAILS-device2, SECURITY-SSHD-6-INFO_GENERAL-device2	→	SECURITY-SSHD-3-ERR_GENERAL-device2
2	IP-DHCPD-3-NOPACKET-device3, IP-TCP-3-BADAUTH-device4, SECURITY-login-4-AUTHEN_FAILED-device4, SECURITY-login-4-AUTHEN_FAILED-ur23-host, SECURITY-SSHD-6-INFO_GENERAL-device2	→	SECURITY-SSHD-3-ERR_GENERAL-device2
3	IP-DHCPD-3-NOPACKET-device3, IP-TCP-3-BADAUTH-device4, SECURITY-login-4-AUTHEN_FAILED-device4, SECURITY-login-4-AUTHEN_FAILED-ur23-host, SECURITY-SSHD-6-INFO_GENERAL-device2	→	SECURITY-SSHD-3-ERR_DETAILS-device2
4	IP-DHCPD-3-NOPACKET-device3, IP-TCP-3-BADAUTH-rr03.host, SECURITY-login-4-AUTHEN_FAILED-device3, SECURITY-login-4-AUTHEN_FAILED-device2, SECURITY-SSHD-6-INFO_GENERAL-device2	→	SECURITY-SSHD-3-ERR_GENERAL-device2
5	IP-DHCPD-3-NOPACKET-device3, IP-TCP-3-BADAUTH-device4, SECURITY-login-4-AUTHEN_FAILED-device4, SECURITY-login-4-AUTHEN_FAILED-ur16-host, SECURITY-SSHD-6-INFO_GENERAL-device2	→	SECURITY-SSHD-3-ERR_GENERAL-device2

Figure 5: Generated Rules through Association Rule Mining

II. These are the results from the generated rules which show the flow of messages on multiple devices: these rules have been considered as potential alarm and configuration changes on NMS system were performed according

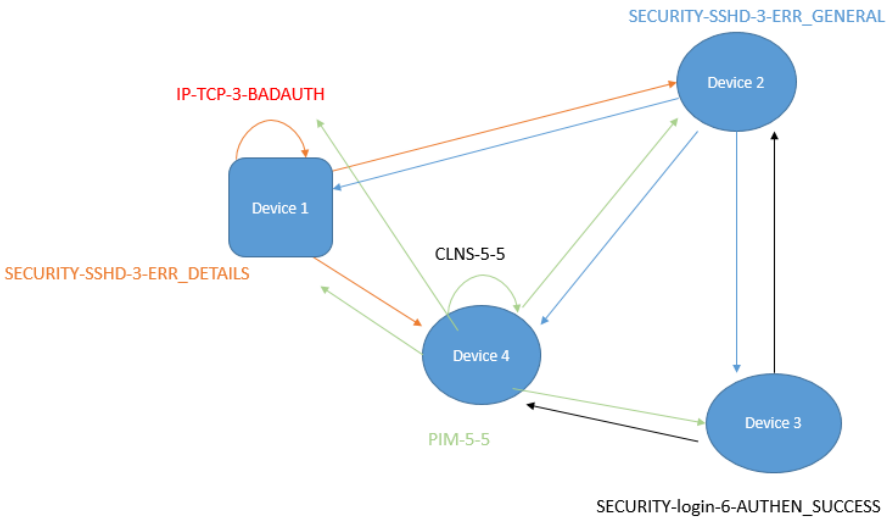


Figure 6: Rules depicting inter-device connectivity

References

[1] WINEPI Approach: https://en.wikipedia.org/wiki/**WINEPI**

[2] Michael Hahsler. A Model-Based Frequency Constraint for Mining Associations from Transaction Data

[3] Michael Hahsler. Mining NB-Frequent Itemsets and NB-Precise Rules

[4] Tongqing Qiu, Jia Wang. What Happened in my Network? Mining Network Events from Router Syslogs

AQI Data – Pune city

IUO (ESRI, Quantela) & Cisco Data Science Team

Data

- API – Provided by Pune City – data can be extracted on a monthly basis.
- AQI data from 50 locations across Pune City available for 22 months starting from Jan'18 – amounts to nearly 1.7M records

- All the pollutants are available multiple times an hour everyday over the given data range

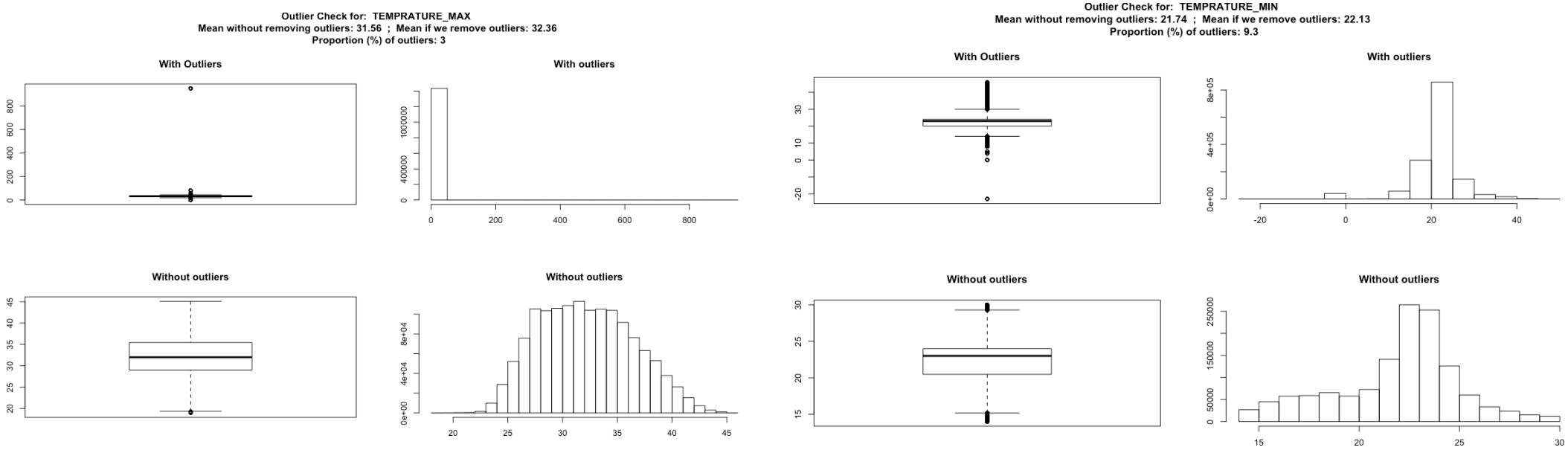
Columns available:

AQI, AQI POLLUTANT, PM10, PM2, CO, CO_2, OZONE, NO, NO_2, SO_2, SOUND, HUMIDITY, LIGHT, AIR PRESSURE, LASTUPDATEDATETIME

- AQI Index has 6 categories as shown below:

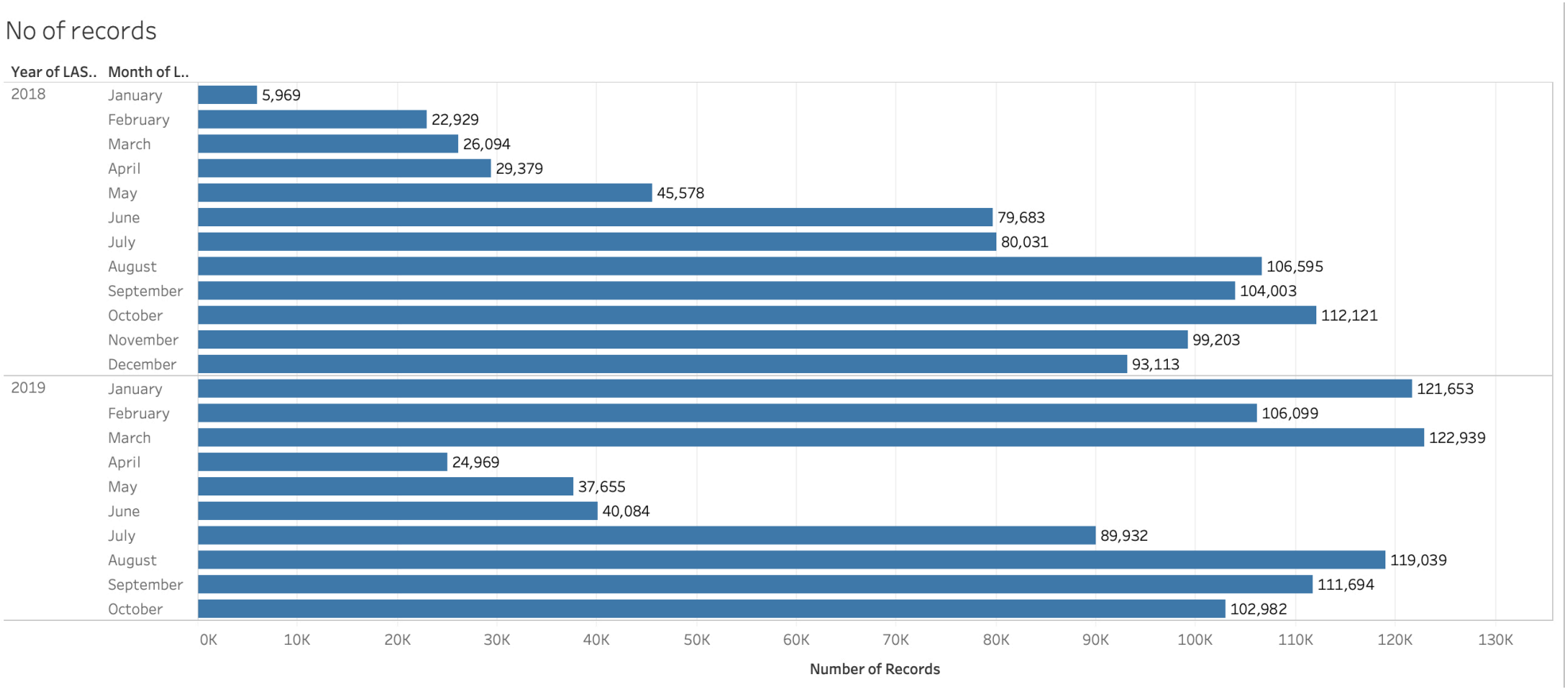
Good (0-50)	Satisfactory (51-100)	Moderately polluted (101-200)	Poor (201-300)	Very poor (301-400)	Severe (> 401)

Data Cleaning & Preprocessing



- Remove Junk data patterns in the variables using Regular Expressions. E.g. Time fields need text cleaning to be machine readable as a Timestamp
- Aggregating at the required level of granularity (hourly, daily)
- Manually obtaining the Latitude / Longitude for all the locations (~50)
- Outlier handling – Removed the exaggerated data points assuming them as erroneous
  - Impute Missing Values – Backward / Forward Filling

Exploratory Analysis



The distribution of the Number of Records by Month is also imbalanced leading to lesser accuracy while forecasting

AQI & Component Features

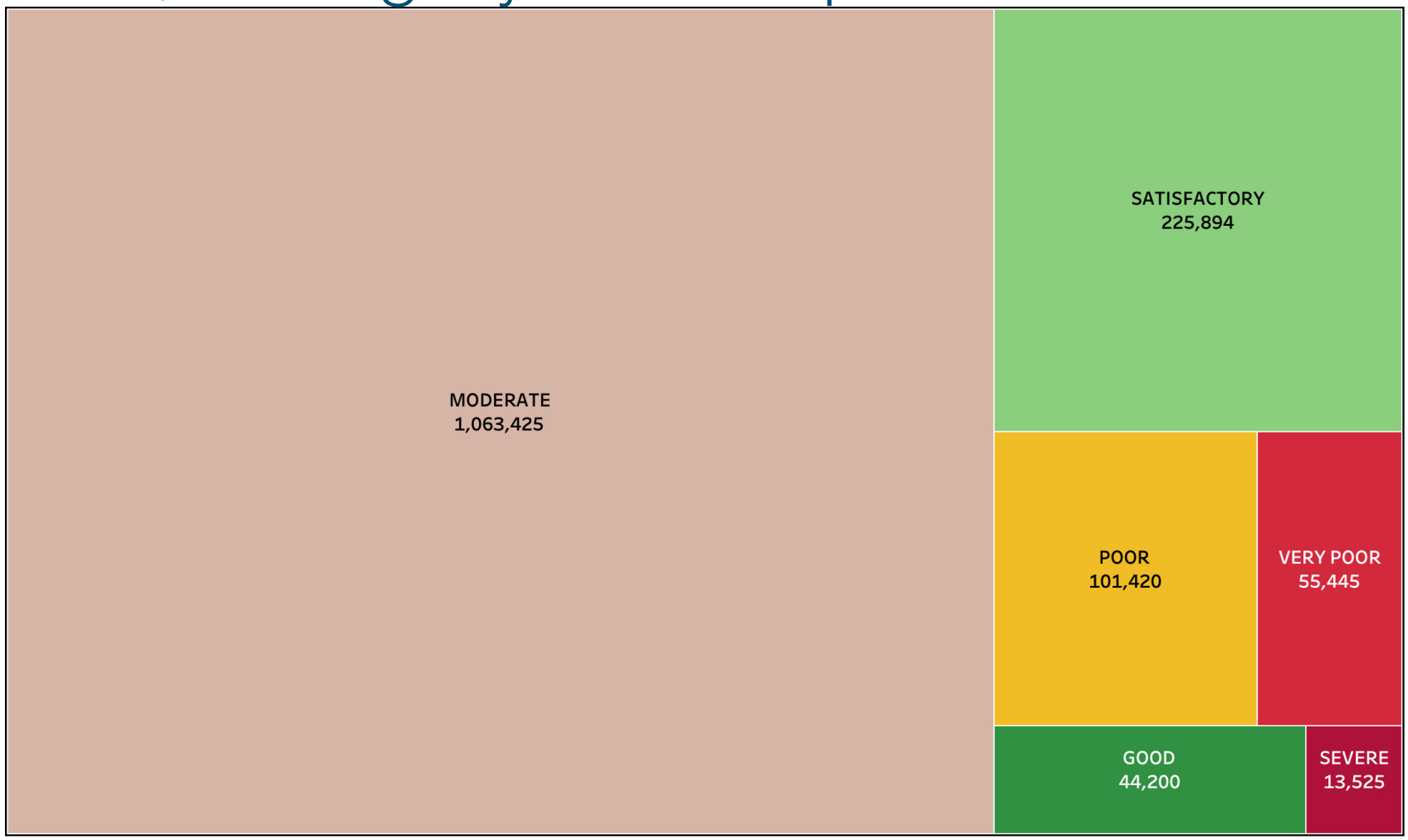
Peak AQI and its component factors are around March, July, November in both the years. Which corresponds with Seasonal changes

These months are coming out significant in a decision tree model as well, further justifying the visual observations.

AQI follows similar pattern as NO_2, OZONE, SO_2

These features have turned out significant in ML models

AQI Category Treemap



Class imbalance

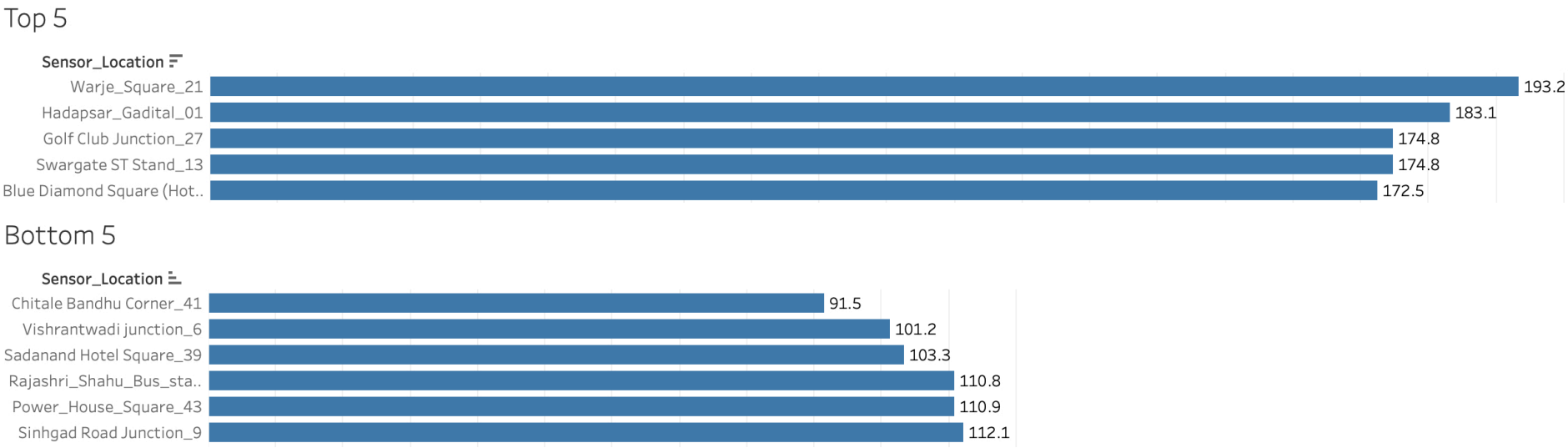
- MODERATE & SATISFACTORY categories cover ~86% of the data
- POOR & VERY POOR account for ~10%
- Remaining 4% is accounted for in GOOD and SEVERE

Hence, the mis-classification errors for the classes “POOR”, “VERY POOR”, “GOOD” & “SEVERE” is expected to be high

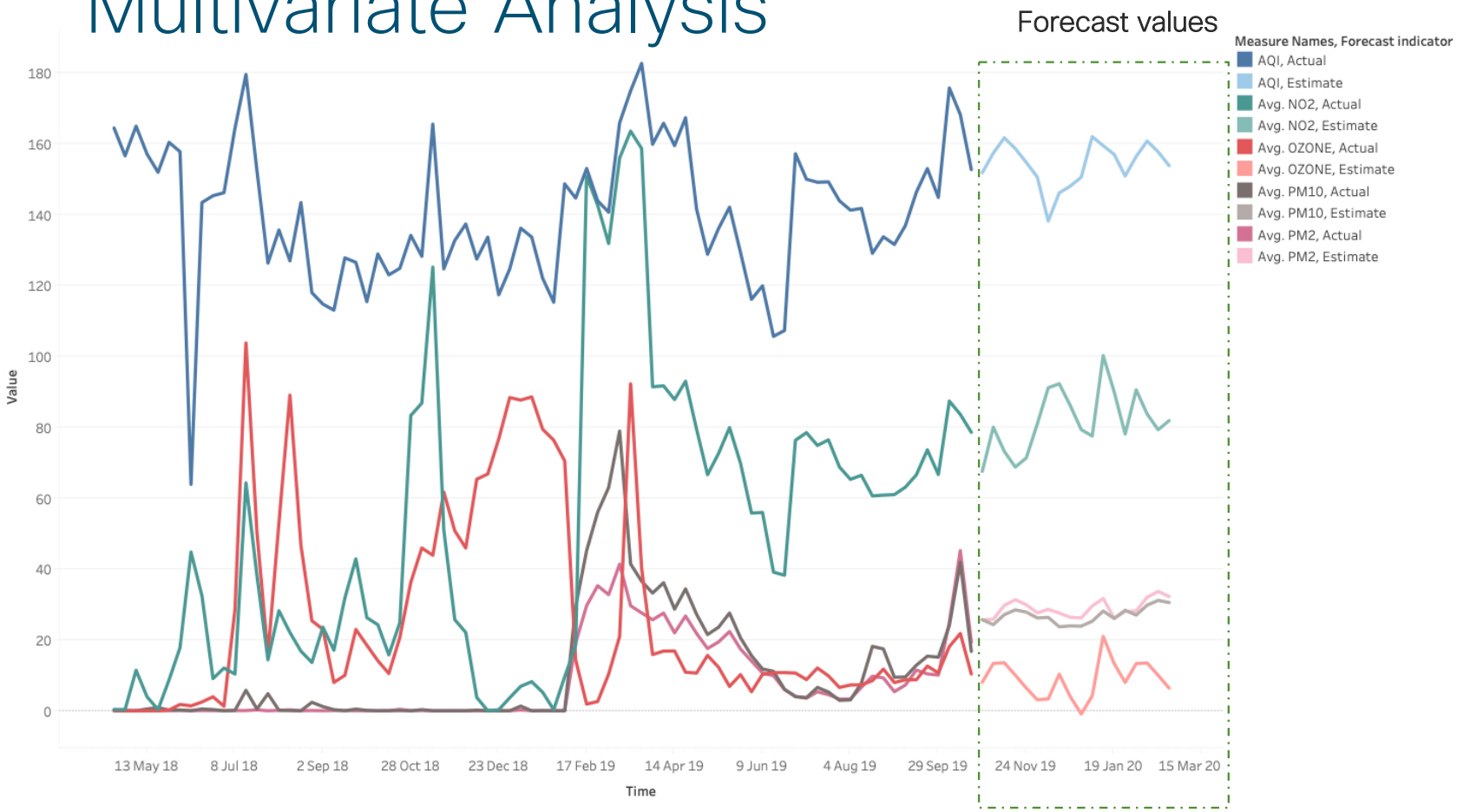
City level snapshot

Average AQI for Pune over the 22 months - 141

There are local variations within the city where the individual AQI Averages vary between 193 and 91

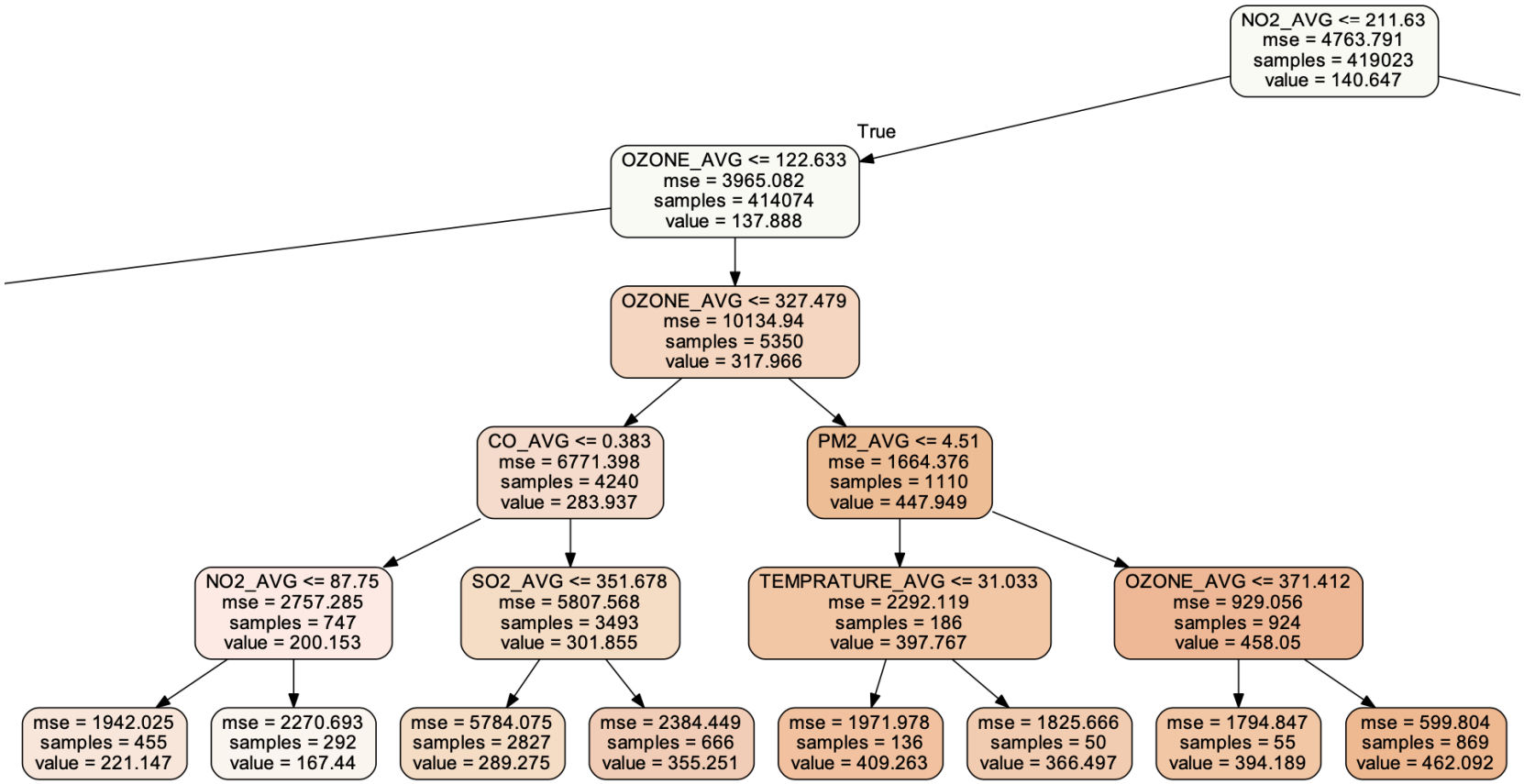


Multivariate Analysis



- A Multivariate time series has more than one time-dependent variable
- Each variable depends not only on its past values but also has some dependency on the other variables. This dependency is used for forecasting future values
- More accurate as correlations are taken in.
- One shot forecast for all variables
- RMSE - 19

Tree Based Models - Regression



- AQI Value is predicted
- Gives the most important features affecting AQI
  - NO_2
  - MONTH
  - OZONE
  - SO_2
  - CO_2
- RMSE ~12

Tree Based Models - Classification

- **AQI Category** is the Target Variable
- Rules generated from the model are easy to interpret
- Helps to deduce breakpoint concentrations of the pollutants for each AQI Category
- Accuracy ~97%

when NO_2 < 26 & DAY >= 194 & OZONE < 21.504 & PM10 < 150 & UV < 0.7 THEN

GOOD

when NO_2 is 191 to 192 & DAY >= 87 THEN

MODERATE

when NO_2 < 55 & OZONE is 47.746 to 124.616 & SO_2 >= 325.8 THEN

POOR

when NO_2 is 192 to 346 & DAY >= 87 THEN

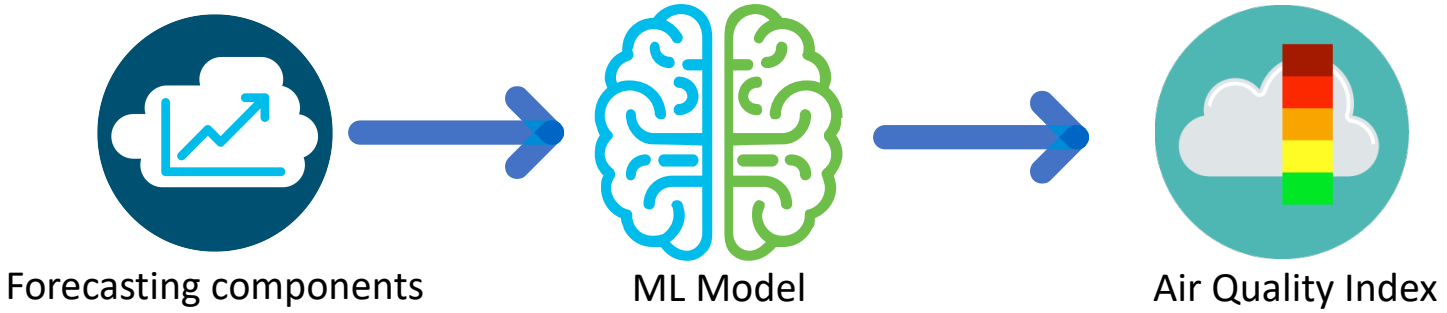
VERY POOR

when NO_2 >= 346 & DAY >= 87 THEN

SEVERE

Application of AQI Modelling

- Timeseries Forecasting models allow the AQI to be predicted into the future (Univariate)
- Filling the gaps – the values from our Forecasting models can be substituted for faulty / missing sensor data to get near precision values
- Important features identified by the models help in effectively tackling the root causes of poor Air Quality



What are the root causes in different areas ?

1. NO_2 : NO_2 gets into the air from burning of fuels. NO_2 forms from emissions from cars, trucks, and buses, power plants, and off-road equipment.

The northern and the north-east part of Pune contains major roads, produces pollutants such as NO2, NO from the vehicles run on petrol and diesel

Warje Square – (proximity to Industrial areas) puts it in the Moderate Category a lot of the time

Pune ST Stand – proximity to a Bus Terminal puts it in the Moderate to Poor Category most of the time

2. SO_2: SO_2 is a toxic gas responsible for the smell of burnt matches. Its released naturally by volcanic activity and is produced as a by-product of copper and the burning of fossil fuels.

Kothrud PMPML Bus Depot has Very high levels of AQI (avg. > 250) presumably by being very close to the Bus Depot

3. Ozone: Ozone is formed, when its precursors (nitrogen oxides and carbon monoxide) generated mainly from fossil fuel combustion react with volatile organic compounds and oxygen in the presence of sunlight.

Shivaji Nagar Chowk & Noble Hospital Square are the most affected by Ozone Pollutant with Average AQIs > 280

What are the root causes in different regions ?

1. The north-west of Pune consist of Industrials area, which make the northern Pune AQI ranges from poor to severe
2. The south-east of Pune city consist of major road junctions and industrials lead to very poor AQI at Fursungikaman and surrounding areas
3. Peak AQI’s and its component factors are high in winter (November & March) and Monsoon (July)