Take-home Exercise 3: Vast Challenge 2025 MC3

CHJ1

2025-06-14

# I am working on this project with Teo Wee Siang Roy.

# I will be working on the Communication Patterns Analysis module of Take-home Exercise 3. This module includes daily communication volume, hourly communication volume, a heatmap of communication patterns based on date and hour, contents of the communcations, and a communication network graph.

# Storyboard of our group’s Shiny application will also be outlined below.

# Load Packages

pacman::p\_load(tidyverse, jsonlite,  
 tidygraph, ggraph,  
 SmartEDA, igraph)

# Importing Data

MC3 <- fromJSON("data/MC3\_graph.json")  
MC3\_schema <- fromJSON("data/MC3\_schema.json")

# Inspect Structure

glimpse(MC3)

List of 5  
 $ directed : logi TRUE  
 $ multigraph: logi FALSE  
 $ graph :List of 4  
 ..$ mode : chr "static"  
 ..$ edge\_default: Named list()  
 ..$ node\_default: Named list()  
 ..$ name : chr "VAST\_MC3\_Knowledge\_Graph"  
 $ nodes :'data.frame': 1159 obs. of 31 variables:  
 ..$ type : chr [1:1159] "Entity" "Entity" "Entity" "Entity" ...  
 ..$ label : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ name : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ sub\_type : chr [1:1159] "Person" "Person" "Person" "Person" ...  
 ..$ id : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ timestamp : chr [1:1159] NA NA NA NA ...  
 ..$ monitoring\_type : chr [1:1159] NA NA NA NA ...  
 ..$ findings : chr [1:1159] NA NA NA NA ...  
 ..$ content : chr [1:1159] NA NA NA NA ...  
 ..$ assessment\_type : chr [1:1159] NA NA NA NA ...  
 ..$ results : chr [1:1159] NA NA NA NA ...  
 ..$ movement\_type : chr [1:1159] NA NA NA NA ...  
 ..$ destination : chr [1:1159] NA NA NA NA ...  
 ..$ enforcement\_type : chr [1:1159] NA NA NA NA ...  
 ..$ outcome : chr [1:1159] NA NA NA NA ...  
 ..$ activity\_type : chr [1:1159] NA NA NA NA ...  
 ..$ participants : int [1:1159] NA NA NA NA NA NA NA NA NA NA ...  
 ..$ thing\_collected :'data.frame': 1159 obs. of 2 variables:  
 .. ..$ type: chr [1:1159] NA NA NA NA ...  
 .. ..$ name: chr [1:1159] NA NA NA NA ...  
 ..$ reference : chr [1:1159] NA NA NA NA ...  
 ..$ date : chr [1:1159] NA NA NA NA ...  
 ..$ time : chr [1:1159] NA NA NA NA ...  
 ..$ friendship\_type : chr [1:1159] NA NA NA NA ...  
 ..$ permission\_type : chr [1:1159] NA NA NA NA ...  
 ..$ start\_date : chr [1:1159] NA NA NA NA ...  
 ..$ end\_date : chr [1:1159] NA NA NA NA ...  
 ..$ report\_type : chr [1:1159] NA NA NA NA ...  
 ..$ submission\_date : chr [1:1159] NA NA NA NA ...  
 ..$ jurisdiction\_type: chr [1:1159] NA NA NA NA ...  
 ..$ authority\_level : chr [1:1159] NA NA NA NA ...  
 ..$ coordination\_type: chr [1:1159] NA NA NA NA ...  
 ..$ operational\_role : chr [1:1159] NA NA NA NA ...  
 $ edges :'data.frame': 3226 obs. of 5 variables:  
 ..$ id : chr [1:3226] "2" "3" "5" "3013" ...  
 ..$ is\_inferred: logi [1:3226] TRUE FALSE TRUE TRUE TRUE TRUE ...  
 ..$ source : chr [1:3226] "Sam" "Sam" "Sam" "Sam" ...  
 ..$ target : chr [1:3226] "Relationship\_Suspicious\_217" "Event\_Communication\_370" "Event\_Assessment\_600" "Relationship\_Colleagues\_430" ...  
 ..$ type : chr [1:3226] NA "sent" NA NA ...

# Extracting the edges and nodes tables

mc3\_nodes <- as\_tibble(MC3$nodes)  
mc3\_edges <- as\_tibble(MC3$edges)

# Cleaning and wrangling nodes

mc3\_nodes\_cleaned <- mc3\_nodes %>%  
 mutate(id = as.character(id)) %>%  
 filter(!is.na(id)) %>%  
 distinct(id, .keep\_all = TRUE) %>%  
 select(-thing\_collected)

mc3\_edges\_cleaned <- mc3\_edges %>%  
 rename(from\_id = source,   
 to\_id = target) %>%  
 mutate(across(c(from\_id, to\_id),   
 as.character)) %>%  
 filter(from\_id %in% mc3\_nodes\_cleaned$id,   
 to\_id %in% mc3\_nodes\_cleaned$id) %>%  
 filter(!is.na(from\_id), !is.na(to\_id))

node\_index\_lookup <- mc3\_nodes\_cleaned %>%  
 mutate(.row\_id = row\_number()) %>%  
 select(id, .row\_id)

mc3\_edges\_indexed <- mc3\_edges\_cleaned %>%  
 left\_join(node\_index\_lookup,   
 by = c("from\_id" = "id")) %>%  
 rename(from = .row\_id) %>%  
 left\_join(node\_index\_lookup,   
 by = c("to\_id" = "id")) %>%  
 rename(to = .row\_id) %>%  
 select(from, to, is\_inferred, type) %>%  
 filter(!is.na(from) & !is.na(to))

used\_node\_indices <- sort(  
 unique(c(mc3\_edges\_indexed$from,   
 mc3\_edges\_indexed$to)))  
  
mc3\_nodes\_final <- mc3\_nodes\_cleaned %>%  
 slice(used\_node\_indices) %>%  
 mutate(new\_index = row\_number())

old\_to\_new\_index <- tibble(  
 old\_index = used\_node\_indices,  
 new\_index = seq\_along(  
 used\_node\_indices))

mc3\_edges\_final <- mc3\_edges\_indexed %>%  
 left\_join(old\_to\_new\_index,   
 by = c("from" = "old\_index")) %>%  
 rename(from\_new = new\_index) %>%  
 left\_join(old\_to\_new\_index,   
 by = c("to" = "old\_index")) %>%  
 rename(to\_new = new\_index) %>%  
 select(from = from\_new, to = to\_new,   
 is\_inferred, type)

# Building the tidygraph object

mc3\_graph <- tbl\_graph(  
 nodes = mc3\_nodes\_final,  
 edges = mc3\_edges\_final,  
 directed = TRUE  
)

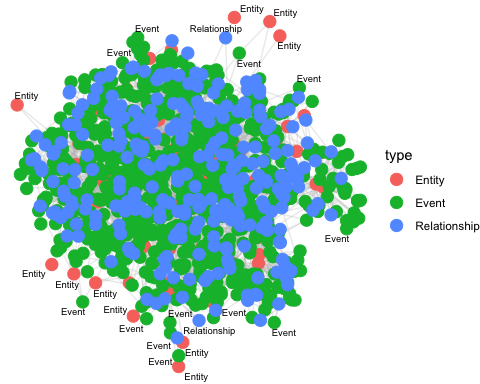
str(mc3\_graph)

Classes 'tbl\_graph', 'igraph' hidden list of 10  
 $ : num 1159  
 $ : logi TRUE  
 $ : num [1:3226] 0 0 0 0 0 0 0 1 1 1 ...  
 $ : num [1:3226] 1137 356 746 894 875 ...  
 $ : NULL  
 $ : NULL  
 $ : NULL  
 $ : NULL  
 $ :List of 4  
 ..$ : num [1:3] 1 0 1  
 ..$ : Named list()  
 ..$ :List of 31  
 .. ..$ type : chr [1:1159] "Entity" "Entity" "Entity" "Entity" ...  
 .. ..$ label : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 .. ..$ name : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 .. ..$ sub\_type : chr [1:1159] "Person" "Person" "Person" "Person" ...  
 .. ..$ id : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 .. ..$ timestamp : chr [1:1159] NA NA NA NA ...  
 .. ..$ monitoring\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ findings : chr [1:1159] NA NA NA NA ...  
 .. ..$ content : chr [1:1159] NA NA NA NA ...  
 .. ..$ assessment\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ results : chr [1:1159] NA NA NA NA ...  
 .. ..$ movement\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ destination : chr [1:1159] NA NA NA NA ...  
 .. ..$ enforcement\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ outcome : chr [1:1159] NA NA NA NA ...  
 .. ..$ activity\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ participants : int [1:1159] NA NA NA NA NA NA NA NA NA NA ...  
 .. ..$ reference : chr [1:1159] NA NA NA NA ...  
 .. ..$ date : chr [1:1159] NA NA NA NA ...  
 .. ..$ time : chr [1:1159] NA NA NA NA ...  
 .. ..$ friendship\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ permission\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ start\_date : chr [1:1159] NA NA NA NA ...  
 .. ..$ end\_date : chr [1:1159] NA NA NA NA ...  
 .. ..$ report\_type : chr [1:1159] NA NA NA NA ...  
 .. ..$ submission\_date : chr [1:1159] NA NA NA NA ...  
 .. ..$ jurisdiction\_type: chr [1:1159] NA NA NA NA ...  
 .. ..$ authority\_level : chr [1:1159] NA NA NA NA ...  
 .. ..$ coordination\_type: chr [1:1159] NA NA NA NA ...  
 .. ..$ operational\_role : chr [1:1159] NA NA NA NA ...  
 .. ..$ new\_index : int [1:1159] 1 2 3 4 5 6 7 8 9 10 ...  
 ..$ :List of 2  
 .. ..$ is\_inferred: logi [1:3226] TRUE FALSE TRUE TRUE TRUE TRUE ...  
 .. ..$ type : chr [1:3226] NA "sent" NA NA ...  
 $ :<environment: 0x10d27e9e8>   
 - attr(\*, "active")= chr "nodes"

# Visualising the knowledge graph

set.seed(1234)

ggraph(mc3\_graph,   
 layout = "fr") +  
 geom\_edge\_link(alpha = 0.3,   
 colour = "gray") +  
 geom\_node\_point(aes(color = `type`),   
 size = 4) +  
 geom\_node\_text(aes(label = type),   
 repel = TRUE,   
 size = 2.5) +  
 theme\_void()



# Extract and Prepare Communication Events

library(lubridate)  
  
# Filter communication events  
comm\_nodes <- mc3\_nodes\_final %>%  
 filter(type == "Event", sub\_type == "Communication") %>%  
 mutate(timestamp = ymd\_hms(timestamp))  
  
# Remove missing timestamps  
comm\_nodes <- comm\_nodes %>% filter(!is.na(timestamp))

# Aggregate Communications by Day and Hour

comm\_nodes <- comm\_nodes %>%  
 mutate(date = as.Date(timestamp),  
 hour = hour(timestamp))

daily\_counts <- comm\_nodes %>%  
 group\_by(date) %>%  
 summarise(comm\_count = n())  
  
hourly\_counts <- comm\_nodes %>%  
 group\_by(date, hour) %>%  
 summarise(comm\_count = n())

# Module for Shiny Application

library(shiny)  
library(ggplot2)  
library(dplyr)  
library(plotly)  
library(lubridate)  
library(DT)  
library(visNetwork)   
  
ui <- fluidPage(  
 titlePanel("Communication Patterns Analysis"),  
 sidebarLayout(  
 sidebarPanel(  
 dateRangeInput("date\_filter", "Date Range:",  
 start = min(comm\_nodes$date),  
 end = max(comm\_nodes$date)),  
 sliderInput("hour\_filter", "Hour Range:",  
 min = 0, max = 23,  
 value = c(0, 23)),  
 width = 3  
 ),  
 mainPanel(  
 tabsetPanel(  
 tabPanel("Daily", plotOutput("daily\_plot")),  
 tabPanel("Hourly", plotOutput("hourly\_plot")),  
 tabPanel("Heatmap", plotlyOutput("heatmap")),  
 tabPanel("Messages", DTOutput("messages\_table")),  
 tabPanel("Network Graph", visNetworkOutput("network\_plot", height = "800px")) # Interactive!  
 ),  
 width = 9  
 )  
 )  
)  
  
server <- function(input, output) {  
 # Filtered communication events by date and hour  
 filtered\_data <- reactive({  
 comm\_nodes %>%  
 filter(date >= input$date\_filter[1],  
 date <= input$date\_filter[2],  
 hour >= input$hour\_filter[1],  
 hour <= input$hour\_filter[2])  
 })  
  
 # Daily Plot  
 output$daily\_plot <- renderPlot({  
 daily\_counts <- filtered\_data() %>%  
 group\_by(date) %>%  
 summarise(comm\_count = n())  
 ggplot(daily\_counts, aes(x = date, y = comm\_count)) +  
 geom\_col(fill = "steelblue") +  
 geom\_text(aes(label = comm\_count), vjust = -0.5, size = 3) +  
 geom\_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +  
 scale\_x\_date(date\_labels = "%Y-%m-%d", expand = expansion(mult = c(0.01, 0.1))) +  
 labs(title = "Daily Communication Volume",  
 x = "Date", y = "Count") +  
 theme\_minimal()  
 })  
  
 # Hourly Plot  
 output$hourly\_plot <- renderPlot({  
 hourly\_totals <- filtered\_data() %>%  
 group\_by(hour) %>%  
 summarise(comm\_count = n())  
 ggplot(hourly\_totals, aes(x = hour, y = comm\_count)) +  
 geom\_col(fill = "steelblue") +  
 geom\_text(aes(label = comm\_count), vjust = -0.5, size = 3) +  
 geom\_smooth(method = "lm", se = FALSE, color = "red", linetype = "dashed") +  
 labs(title = "Hourly Communication Volume",  
 x = "Hour", y = "Count") +  
 theme\_minimal()  
 })  
  
 # Heatmap  
 output$heatmap <- renderPlotly({  
 hourly\_counts <- filtered\_data() %>%  
 group\_by(date, hour) %>%  
 summarise(comm\_count = n(), .groups = "drop")  
 p <- ggplot(hourly\_counts, aes(  
 x = hour, y = date, fill = comm\_count,  
 text = paste("Date:", date,  
 "<br>Hour:", hour,  
 "<br>Communications:", comm\_count))) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient(low = "white", high = "firebrick") +  
 scale\_y\_date(date\_labels = "%Y-%m-%d") +  
 labs(title = "Hourly Communication Patterns",  
 x = "Hour", y = "Date") +  
 theme\_minimal()  
 ggplotly(p, tooltip = "text")  
 })  
  
 # Messages Table  
 output$messages\_table <- renderDT({  
 filtered\_data() %>%  
 arrange(desc(timestamp)) %>%  
 mutate(Time = format(timestamp, "%Y-%m-%d %H:%M:%S")) %>%  
 select(Time, Content = content)  
 }, options = list(  
 pageLength = 10,  
 autoWidth = TRUE,  
 scrollX = TRUE,  
 columnDefs = list(  
 list(width = '20%', targets = 0),  
 list(width = '80%', targets = 1)  
 )  
 ),  
 rownames = FALSE  
 )  
  
 # Interactive Network Graph  
 output$network\_plot <- renderVisNetwork({  
 req(filtered\_data())  
 # Get nodes in filtered timeframe  
 peak\_comm\_ids <- filtered\_data() %>% pull(new\_index)  
 # Get edges involving these nodes  
 peak\_edges <- mc3\_edges\_final %>%  
 filter(from %in% peak\_comm\_ids | to %in% peak\_comm\_ids)  
 # Get unique node indices from edges  
 used\_indices <- unique(c(peak\_edges$from, peak\_edges$to))  
 # Create filtered node table  
 peak\_nodes\_sub <- mc3\_nodes\_final %>%  
 filter(new\_index %in% used\_indices) %>%  
 mutate(sub\_index = row\_number())  
 # Update edge indices  
 peak\_edges\_sub <- peak\_edges %>%  
 left\_join(select(peak\_nodes\_sub, old\_index = new\_index, from\_sub = sub\_index),   
 by = c("from" = "old\_index")) %>%  
 left\_join(select(peak\_nodes\_sub, old\_index = new\_index, to\_sub = sub\_index),   
 by = c("to" = "old\_index")) %>%  
 select(from = from\_sub, to = to\_sub, is\_inferred, type) %>%  
 filter(!is.na(from) & !is.na(to)) %>%  
 mutate(id = row\_number(),  
 arrows = ifelse(is\_inferred, "to;from", "to"),  
 dashes = is\_inferred,  
 title = paste("Edge type:", type))  
 # Prepare nodes for visNetwork  
 nodes\_vis <- peak\_nodes\_sub %>%  
 mutate(  
 id = sub\_index,  
 label = paste0(label, "\n(", type, ")"),  
 group = type,  
 title = paste0(  
 "Name: ", label,   
 "\nType: ", type,  
 if ("sub\_type" %in% names(.)) paste0("\nSubtype: ", sub\_type) else "",  
 "\nConnections: ", purrr::map\_dbl(sub\_index, ~sum(peak\_edges\_sub$from == .x | peak\_edges\_sub$to == .x))  
 )  
 )  
 # visNetwork plot  
 visNetwork(nodes\_vis, peak\_edges\_sub, height = "800px", width = "100%") %>%  
 visOptions(  
 highlightNearest = list(enabled = TRUE, degree = 2, hover = TRUE),  
 nodesIdSelection = TRUE,  
 selectedBy = list(variable = "group", multiple = TRUE)  
 ) %>%  
 visPhysics(stabilization = TRUE) %>%  
 visInteraction(navigationButtons = TRUE) %>%  
 visLegend()  
 })  
}  
  
shinyApp(ui, server)

# Storyboard of Shiny Application

The aim of our Shiny application is to identify suspicious communication patterns.

The combination of the daily communication volume bar chart, hourly communication volume bar chart, and the hourly heat map is one way to detect temporal anomalies and shifts in communication behaviour.

Daily Communication Volume Bar Chart

This chart displays the number of communications per day, overlaid with a red dashed trend line. It allows analysts to quickly spot days with unusually high or low activity, as well as to observe overall trends such as gradual increases or decreases in daily volume. A sudden spike or drop may indicate an event or coordinated action that warrants further attention.

Hourly Communication Volume Bar Chart

By aggregating the number of communications by hour across the dataset and plotting a trend line, this chart reveals the typical daily rhythm of communications. Significant deviations from the expected hourly pattern, such as an unusual surge in late-night or early-morning activity, can be a red flag for out-of-pattern behaviour.

Hourly Communication Heat Map

The heat map visualises the intensity of communications for each hour of each day, making it easy to spot recurring peaks or abrupt changes. Consistent high activity during certain hours may be normal, but a shift in these patterns, such as a new peak at an odd hour or an isolated day with unusual activity, can signal suspicious coordination or attempts to evade detection.

Together, these temporal visualisations enable analysts to:

1. Detect when communication patterns deviate from the established norm.
2. Pinpoint specific dates and hours where suspicious activity is concentrated.
3. Observe whether suspicious patterns are isolated incidents or part of a broader trend.

Communication Network Graph

The communication network graph provides a structural snapshot of the network at any selected date and time. By filtering the network to a specific window (identified as suspicious via the temporal charts), analysts can:

1. Visualise which entities are most active or central during periods of interest.
2. Identify clusters, bridges, or unusually dense connections that may indicate coordinated groups or key facilitators.
3. Assess the scale and reach of the network at critical moments, helping to distinguish between routine communication and orchestrated events.

This graph is essential for understanding not just when, but who is involved in suspicious activity, and how information or influence flows between individuals and organisations.

A slicer has been incorporated on the left side of the dashboard, allowing users to filter all visualisations by date and time. This interactive control enables analysts to dynamically adjust the temporal scope of the data displayed, making it easier to focus on specific periods of interest and observe how communication patterns and network structures change over time

Deep Dive: The Messages Tab

Once a suspicious period is identified through the above visualisations, the messages tab allows for targeted review of the actual message content exchanged during that window. This enables:

1. Direct examination of the nature and context of communications, confirming whether the content aligns with legitimate business or hints at illicit coordination.
2. Identification of code words, unusual instructions, or references to sensitive operations.
3. Verification of relationships and roles among key entities, as messages often reveal hierarchy, influence, and intent.

By correlating message content with network structure and temporal anomalies, analysts can move from pattern detection to concrete evidence gathering, supporting investigations into illegal or unethical activities.

Cluster

The visualisation reveals the complex web of relationships in the Oceanus maritime environment around Nemo Reef. This network map serves as a digital detective board, showing how 33 key entities, both people and vessels, communicate and interact in this sensitive marine area.

The Communication Patterns:

Central Hub Activity: Kelly and Sam emerge as information brokers, maintaining connections across multiple groups, potentially serving as intermediaries between legitimate and suspicious operations.

Isolated Operations: Some entities like Northern Light and Mariner’s Dream operate with limited connections, suggesting either independent operations or intentional isolation.

Bridge Connections: Certain individuals like Nadia Conti and Liam Thorne serve as critical bridges between official channels and operational networks.

Cluster Analysis

The dashboard presents a comparative analysis of different clustering algorithms applied to the Oceanus maritime network, aiming to uncover meaningful communities among people and vessels involved around Nemo Reef.

The Entity Communication Network reveals that Nemo Reef operations exist within a complex ecosystem where environmental protection, legitimate tourism, and potentially suspicious activities intersect. The visualisation provides maritime authorities with a strategic overview to prioritise investigations, enhance monitoring of key relationships, and protect both marine conservation efforts and legitimate maritime commerce while identifying threats to the reef’s ecological integrity.

The story this network tells: While legitimate conservation and tourism operations maintain their protective roles around Nemo Reef, a concerning undercurrent of potentially coordinated suspicious activities requires immediate investigative attention to preserve both maritime security and environmental protection in these critical waters.