Journal of Mathematics and Computing Science, 12 (1): 41-54, 2026
Copyright © UiTM Press
eISSN: 0128-0767

GutVision: Data Visualisation of Digestive Disorders in Malaysia

Siti Aishah Mohmad Saaid[1], Norisan Abd Karim [2]*, Fariza Hanis Abdul Razak [3], Nor Diana Ahmad[4], Fauziah Redzuan[5] and Rogayah Abdul Majid[6]

[1,2,3,4,5,6]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia

Authors' email: aishahsaaid24@gmail.com, norisanak@uitm.edu.my*, fariza062@uitm.edu.my, nordiana@tmsk.uitm.edu.my , fauziahr@tmsk.uitm.edu.my , rogayah@tmsk.uitm.edu.my

*Corresponding author

Received 23 February 2026; Received in revised 30 April 2026; Accepted 21 May 2026
Available online 15 June 2026
DOI: https://doi.org/10.24191/0g0pj532

Abstract: Digestive disorders are on the rise in Malaysia, increasing healthcare costs and affecting diverse population groups. Current monitoring systems are only able to offer localised insights. This paper presents GutVision, an interactive dashboard that explores 151,268 rows of digestive health information (2019-2023) datasets from Pusat Informatik Kesihatan, which were cleaned to 149,750 using Python-based ETL pipelines for data processing. The project is aimed at illnesses like irritable bowel syndrome (IBS) and inflammatory bowel disease (IBD) and uses descriptive analysis to emphasise demographic and geographical variations. Results show Johor, Selangor and Sabah had the highest admissions, with 21,560, 17,200 and 14,666 admissions, and the highest digestive disorders recorded are unspecified gastroenteritis with 47,712 cases, followed by malignant neoplasm of the colon with 39,527 cases and 10,557 cases of gastro-oesophageal reflux disease. GutVision was created on a big data platform using Agile and OSEMN processes, Python to clean data and Power BI to visualise it. The dashboard has filters of age, gender, state and year, and it is possible to explore it meaningfully. GutVision increases knowledge, decision-making, and healthcare planning in Malaysia by linking health information with a user-friendly design.

Keywords: Big Data, Dashboard, Digestive Disorders, Data Visualisation, Malaysia

1 Introduction

In Malaysia, the rate of digestive disorders, including irritable bowel syndrome, inflammatory bowel disease (IBD), and colorectal cancer, has been on the increase, requiring rapid attention as a public health issue. Alarmingly, the rates of late-stage diagnosis of colorectal cancer are 66% of females and 54% of males, which is a strong indication of the necessity of early intervention [1]. The use of technologies like dashboards and predictive indices has been embraced in other countries but not in Malaysia, which reduces the possibility of detecting the trends and monitoring the digestive health. Existing Ministry of Health Malaysia data presented through the KKMNOW Data Catalogue such as infant immunisation coverage, blood donation activities and PeKa B40 health screenings are focused on broad indicators and program-level data instead of disease-specific analytics [2]. Unlike these reports, GutVision enables granular, interactive analysis of digestive disorders across demographics, providing analytical depth. This gap affects the quality of care and its timeliness for patients who have chronic gastritis and other related issues. Better visualisation tools would help strengthen the collection and analysis of data that could facilitate more specific health interventions and early detection,

This is an open access article under the CC BY-SA license (https://creativecommons.org/licenses/by-sa/3.0/).

increasing public awareness [3-4]. This technological gap must be overcome to lower the increasing load of digestive disorders in Malaysia.

To overcome these problems, the GutVision project creates an interactive Power BI dashboard, a tool that enables individuals to examine the data on digestive disorders in Malaysia using maps, charts, and tables. The project scope would include determining the data visualisation requirements, designing the appropriate visualisation models, and developing the GutVision dashboard. The proposed solution will enhance the knowledge of digestive disorder trends and assist in healthcare planning and policy decision-making. With data visualisation using Python and Power BI, GutVision can be useful to address the problem of dealing with and understanding data regarding health. Transforming raw data into interactive dashboards brings users to investigate demographic and regional patterns in detail and make decisions about awareness campaigns and early diagnosis, as well as how to manage digestive disorders in Malaysia.

A Big Data and the Characteristics

The requirements made by big data in healthcare have the essential characteristics outlined by prior researchers as volume, variety, velocity, veracity, and value, as well as variability [5]. The use of big datasets in medicine and healthcare encounters key issues depending on a few factors that enhance complexity in the review and evaluation procedure. Volume refers to the large amount of healthcare information that is collected via electronic health records (EHRs), medical imaging technology, or the use of wearable devices. This project processed around 150,000 patient records from 2019-2023. Patient information needs enhanced data storage and processing mechanisms to work with them [6]. Variety is related to the structured (numerical lab results), semi-structured (XML files), and unstructured (clinical notes, images) types of data in healthcare. The dataset used in this study contained structured data, including demographic information, admission details, and diagnosis codes. Healthcare data of different kinds requires machine learning combined with natural language processing (NLP) so that it will be usable [7].

Velocity refers to the speed at which medical data is acquired and used, particularly in real time, such as continuously monitoring a patient or the citation of a disease outbreak. For this project, data was processed in batches from yearly hospital records. Fast processing of data is a key issue in the decision-making of effective healthcare and applying measures in time [8]. Veracity addresses the reliability and validity of the data because low data quality will always skew many of the healthcare insights. For this project, data cleaning removed around 1,500 records with missing values. Such practices as data cleansing and statistical validation increase the reliability of the analytical findings [6]. Value describes how big data is used to inform actionable data to improve patient care and make decisions about the health of the population. GutVision provides actionable insights for healthcare resource planning. Advanced algorithms help to determine why one is making an evidence-based decision [9]. Finally, variability represents the lack of disparity in many aspects, including time intervals, data formats, measurement procedures, and quality, which presents a challenge to integrating data. It is also vital to have standardisation frameworks that need to be in place to harmonise data that are heterogeneous [10]. For this project, data standardisation was required due to inconsistent formatting across different years.

B Big Data Visualisation

In healthcare analytics, big data visualisation plays an important role in converting complex data sets into actionable information. When conducting research on digestive disorders in Malaysia, the tools provide a systematic method of studying the prevalence of a disorder and other demographic factors so that policymakers, researchers, and care providers make informed decisions. Tools such as Power BI can support and enable large-dataset analysis and interactive data exploration through drilling down and filtering. A challenge encountered was ensuring query performance was smooth when applying multiple

filters simultaneously to the entire dataset. Power BI handles large datasets through in-memory analytics and data compression, even though we encountered performance degradation when applying multiple concurrent filters to the full 149,750 record dataset. Prior researchers [8] state that public health surveillance dashboards provide the ability to combine several different data sources to create real-time monitoring need immediate attention and have tremendous value. Prior researchers observed that besides the usability of the healthcare dashboards, another graduate attribute is the interactivity and information display that allows the use of these dashboards in the public health sector of epidemiology and managing disorders [11].

2 Material and Methods

Figure 1 shows the OSEMN and Agile methodology. The OSEMN method and the Agile techniques are useful for analysing and visualising information, as they offer both structure and flexibility in developing dashboards for digestive disorders. The initial dataset contained 151,268 rows with 15 columns, including patient demographics, diagnosis information, and admission details. Data quality assessment showed that two columns had over 98% missing values and were removed. The OSEMN streamlines data science tasks, whereas the Agile methodology enables an iterative cycle of refinements, especially when working with large clinical datasets, whereby conclusions vary with continuous data processing, modelling, and visualisation. Agile also has the advantage in data-intensive projects because it takes inputs from users to deliver improved results [12] yet does not solve the breaches in data science processes entirely. OSEMN, on the other hand, is a systematic approach to data acquisition, scrubbing, discovery, modelling, and interpretation [13], but it can be constrained in a demanding environment where data may have to be frequently updated, as in the case of interactive dashboards. A combination of these two frameworks offers flexibility and accuracy by synchronising OSEMN steps with iterative refinement cycles of Agile. The combination of the OSEMN-structured decision-making process and the Agile adapting approach facilitates the creation of reliable user-centred solutions for data visualisation. The interrelation, as stated in Table 1, is as follows:

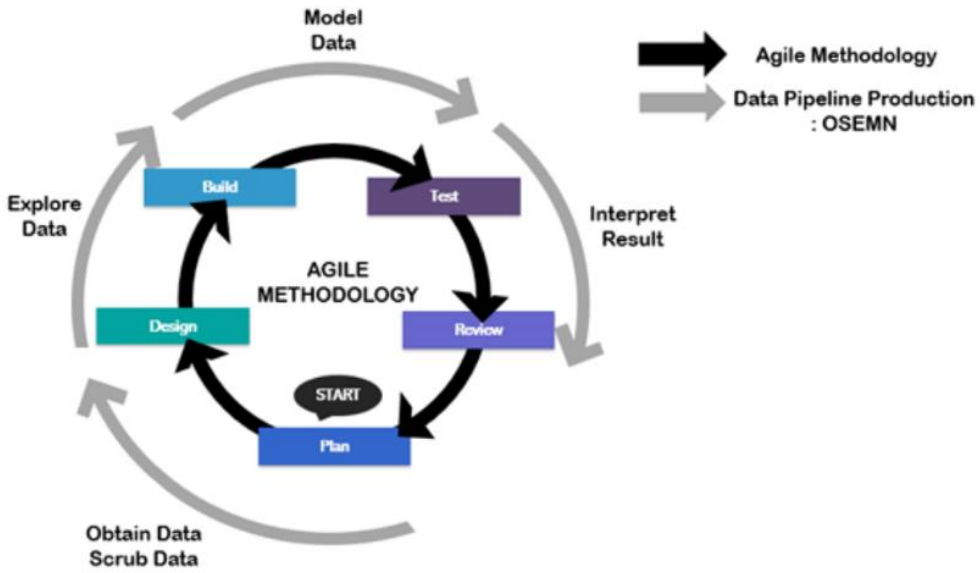

Figure 1: Shows the Agile and OSEMN Model Methodology

Table 1: Methodology Phases

Agile	OSEMN	Description
Plan	Obtain Data (Identification of data requirement)	Identification of datasets of digestive disorder (2019-2023) from Pusat Informatik Kesihatan, Ministry of Health Malaysia. The type of disorder selected: the demographics (age, gender, state) and year are set as the data requirements.
	Scrub Data (Data Cleaning)	Cleaning raw data with Python, including missing values, finding duplicates along with formatting that provides consistent inputs into visualisations
Design	Explore (Storyboard Design)	Designing with storyboard and visualisation layout of GutVision dashboard. Determining effective visual images (maps, trend lines, bar graphs, pie charts) to use that show disorder prevalence and demographic breakdown.
Build	Model (Prototype of Data Visualization)	Developing dashboard in Power BI on clean dataset. Including interactive options, filters (by year, facility state and disorder type).
Test	Interpret Result	Test dashboard functions to be accurate and easy to use. Analysing data to make them fit in patterns (e.g., gastritis prevalence, state comparisons) and represent them in an understandable way.
Review		Reviewing and presenting GutVision dashboard to supervisor and users and optimising design with their suggestions. Some changes will involve making visualisations more transparent, enabling drill-down, and making the data readable to users.

A Visualisation Process

The GutVision project offers a systematic analysis of digestive disorders in Malaysia in four methods: data collection from Pusat Informatik Kesihatan; processing where the final cleaned dataset is 149,750 rows after the removal of 2 columns with >98% missing data, design, and dashboard development using Power BI. The sources of information include secondary databases provided by the Ministry of Health, specific to IBS, IBD, and colorectal cancer, and demographic and geographical data, including age, gender, ethnicity, and state. Asgari and Hurtut explain that structured storyboarding can represent the components and narrative structure of data-driven stories, supporting the prototyping process and visual communication [14]. True to this, GutVision makes use of these demographic and diagnostic factors to have useful comparisons. Analysis revealed key patterns: senior patients (65+) represent the largest age group (23.08%); admissions were highest in Johor (21,560), Selangor (17,200), and Sabah (14,666); and 11,675 outliers of total admission days were filtered. Huisman points out that thorough collection of data can provide a solid foundation of verifiable hypothesis testing and solid research [15].

The data was cleaned, organised, and standardised in Python to guarantee precision because disorganised information can be misinterpreted [16]. Disorders were grouped into clinical considerations; on the other hand, missing values were labelled as gaps. The storyboarding and visual presentation were aimed at making it more readable and engaging, while the structured visual narrative helps connect different visualisation components and communicate information more effectively [14-15]. Lastly, the cleaned data were placed into an interactive Power BI dashboard where users could filter by year, state, age group, gender, and disorder type with visible representations, including line and bar charts, to see the trends. At the end of this structured procedure, GutVision can present evidence-based insights to assist evidence-based policymaking and population health plans in the management of digestive disease in Malaysia.

B Data Cleaning for GutVision Development

This part will concentrate on data cleaning and processing in developing GutVision. Python was used to ensure that the data was accurate, consistent, and easy to work with and could then be used to visualise and analyse it. The details of the data quality summary are provided in Table 2, as follows:

Table 2: Data Quality Summary Before and After Cleaning

Data Quality Metrics	Before Cleaning	After Cleaning
Total Rows	151,268	149,750
Total Columns	15	14
Records with Missing Key Data	1,518	0
Columns with >98% Missing	2 columns	0
Data completeness Ratio	99%	

```
import pandas as pd
import os
```

```
#Upload Files:
# List of file paths
file_paths = [
    'Inp2019_digestive.xlsx',
    'Inp2020_digestive.xlsx',
    'Inp2021_digestive.xlsx',
    'Inp2022_digestive.xlsx',
    'Inp2023_digestive.xlsx'
]

# Load all files into a list of dataframes
df_list = [pd.read_excel(file_path) for file_path in file_paths]

# Optionally, add a 'Year' column to each dataframe (assuming year is part of the filename)
for i, file_path in enumerate(file_paths):
    year = file_path.split('Inp')[1].split('_')[0]  # Extract year from the filename
    df_list[i]['Year'] = int(year)  # Add a 'Year' column to each dataframe
```

```
#Merge DataFrames
# Merge all the files into one dataframe
merged_df = pd.concat(df_list, ignore_index=True)

# Verify the shape of the merged dataframe
print(f"Total rows and columns after merging: {merged_df.shape}")
```

```
Total rows and columns after merging: (151268, 15)
```

Figure 2: Loading and Merging Data

Figure 2 shows the merging of the 2019-2023 datasets is a significant step in the entire data-cleaning process. To implement this step, each separate Excel file will be read into an individual dataframe using the Pandas library, as shown in the accompanying figure. All the files are read into a data frame with the read excel functionality, and a Year column is added by extracting the year information inherent in the filenames. This labelling process prevents temporal inaccuracy in each record. After they are enriched with the year identifier on an individual data frame, pd.concat() is run to combine all of them into a single and comprehensive data frame. This kind of unified structure leads to consistency, thus simplifying subsequent trend analyses.

```
# Calculate the count of missing values in each column
missing_rows_count = merged_df.isnull().sum()

# Display columns with missing values and their row counts (sorted by missing rows)
print("\nColumns with missing values (count of missing rows):")
print(missing_rows_count[missing_rows_count > 0].sort_values(ascending=False))
```

```
Columns with missing values (count of missing rows):
ucod_icd10_code    149074
ucod_icd10_desc    149074
dtype: int64
```

```
# Calculate the count of missing values for each column
missing_rows_count = merged_df.isnull().sum()

# Identify columns with more than 98% missing values (threshold: 2% non-missing data)
columns_to_drop = missing_rows_count[missing_rows_count > len(merged_df) * 0.98].index

# Drop columns with >98% missing values
df_cleaned = merged_df.drop(columns=columns_to_drop)

# Display the columns remaining after dropping
print("Columns remaining after dropping columns with >98% missing values:")
print(df_cleaned.columns.tolist())

# Show the columns that were dropped
print("\nColumns dropped due to >98% missing values:")
print(columns_to_drop.tolist())

# Verify the shape after dropping columns
print(f"\nOriginal DataFrame shape: {merged_df.shape}")
print(f"DataFrame shape after dropping columns with >98% missing values: {df_cleaned.shape}")
```

```
Columns remaining after dropping columns with >98% missing values:
['date_adm_disch_year', 'date_admission_full_date_desc', 'date_discharge_full_date_desc', 'facility_state', 'facility_district', 'patient_state', 'patie
nt_age_single', 'patient_gender', 'patient_ethnicity_name', 'disch_type_desc', 'icd10_code', 'icd10_desc', 'Year']

Columns dropped due to >98% missing values:
['ucod_icd10_code', 'ucod_icd10_desc']

Original DataFrame shape: (151268, 15)
DataFrame shape after dropping columns with >98% missing values: (151268, 13)
```

Figure 3: Dropping Columns with Excessive Missing Values

Figure 3 shows that data cleaning is followed by the identification and deletion of two columns that had over 98% missing values. Columns with values lower than this threshold were removed since they did not have sufficient data to power meaningful insights; thus, removal of such columns provides further focus on relevant variables in the dataset. The removal is done by the drop() method, and this improves the quality and reliability of what is left so that it can be further explored. To make sure that no valuable data will be accidentally deleted, the form of the data frame is verified before and after this process of cleaning to make sure that only informative columns will come out to be visualised and analysed.

Figure 4 demonstrates the actions taken to clean the data – by replacing placeholders, which are covariate markers that show missing/unavailable information such as ‘NOT AVAILABLE’, Tiada Maklumat’ and ‘RAGU’ are converted to NaN, and columns with a very high proportion of missing data (over 98%) are dropped. The missing values of the attribute’s patient_state, patient_gender, patient_ethnicity_name, and disch_type_desc after the replacement procedure constitute 93134, 275, 1118, and 136, respectively. The figures reflect the extent of data that was, in actual sense, never present in some of the specified columns even after the replacement was done. These missing values are evidence of gaps in the data that may require more attention or adoption of better imputation techniques to conduct effective analysis. A replacement of NaN would preserve a similar pattern of missing values, and dropping or deleting columns that contain too much missing data would improve the quality and feasibility of the data utilised.

```python
import pandas as pd

# List of values to replace with NaN
replace_values = ['NOT AVAILABLE', 'Tiada Maklumat', 'RAGU']

# Display the original shape (rows, columns) of the merged dataframe
print(f"Original DataFrame shape: {merged_df.shape}")

# Loop through the specified columns and replace placeholders with NaN
for col in ['patient_state', 'patient_gender', 'patient_ethnicity_name', 'disch_type_desc']:
    # Replace the placeholder values with NaN
    merged_df[col] = merged_df[col].replace(replace_values, pd.NA)

    # Calculate and print the count of missing values (NaN) for each column after replacement
    missing_count = merged_df[col].isna().sum()
    print(f"Missing values in '{col}' after replacement: {missing_count}")

# Display the shape of the dataframe after replacing placeholders
print(f"\nDataFrame shape after replacing placeholders with NaN: {merged_df.shape}")

# Clean column names by stripping any leading or trailing spaces
merged_df.columns = merged_df.columns.str.strip()

# Drop columns where more than 98% of the values are missing
df_cleaned = merged_df.dropna(thresh=len(merged_df) * 0.02, axis=1)

# Check columns before and after dropping
print(f"Columns before dropping: {merged_df.columns.tolist()}")
print(f"Columns after dropping: {df_cleaned.columns.tolist()}")

# Display the final shape of the dataframe after dropping columns
print(f"DataFrame shape after dropping columns with >98% missing values: {df_cleaned.shape}")
```

```
Original DataFrame shape: (151268, 15)
Missing values in 'patient_state' after replacement: 93134
Missing values in 'patient_gender' after replacement: 275
Missing values in 'patient_ethnicity_name' after replacement: 1118
Missing values in 'disch_type_desc' after replacement: 136

DataFrame shape after replacing placeholders with NaN: (151268, 15)
Columns before dropping: ['date_adm_disch_year', 'date_admission_full_date_desc', 'date_discharge_full_date_desc', 'facility_state', 'facility_district', 'patient_state', 'patient_age_single', 'patient_gender', 'patient_ethnicity_name', 'disch_type_desc', 'icd10_code', 'icd10_desc', 'ucod_icd10_code', 'ucod_icd10_desc', 'Year']
Columns after dropping: ['date_adm_disch_year', 'date_admission_full_date_desc', 'date_discharge_full_date_desc', 'facility_state', 'facility_district', 'patient_state', 'patient_age_single', 'patient_gender', 'patient_ethnicity_name', 'disch_type_desc', 'icd10_code', 'icd10_desc', 'Year']
DataFrame shape after dropping columns with >98% missing values: (151268, 13)
```

Figure 4: Replacing Placeholders and Dropping Columns with Excessive Missing Data

```python
# Drop rows where any of the relevant columns have missing values
df_cleaned = df_cleaned.dropna(subset=['patient_gender', 'patient_ethnicity_name', 'disch_type_desc'])

# Show shape after dropping
rows, cols = df_cleaned.shape
print(f"Data shape after cleaning: {rows} rows × {cols} columns")
```

```
Data shape after cleaning: 149766 rows × 13 columns
```

```python
# Check for any remaining nulls in those key columns
print(df_cleaned[['patient_gender', 'patient_ethnicity_name', 'disch_type_desc']].isna().sum())
```

```
patient_gender           0
patient_ethnicity_name   0
disch_type_desc          0
dtype: int64
```

Figure 5: Dropping Rows with Missing Values in Key Columns

Figure 5 shows that the next stage is deleting the rows with filled-in missing values in the key columns disch_type_desc, patient_ethnicity_name, and patient_gender in any of these blank areas when the total row is removed in the data set. After this cleaning, there is a follow-up on any remnant holes in the same columns as a blanket check on the quality of data to be further analysed. This data separates the records into categories according to age, geographical region, etc. in case such a study is being done on digestive health trends. A missing element in any of such crucial areas can bias the findings in the opposite direction, and so one of the ways to come up with cleaner and more credible results is by cutting out the blank ones.

```
#Convert string dates to pandas datetime format
merged_df['date_admission_full_date_desc'] = pd.to_datetime(merged_df['date_admission_full_date_desc'], errors='coerce')
merged_df['date_discharge_full_date_desc'] = pd.to_datetime(merged_df['date_discharge_full_date_desc'], errors='coerce')
```

Figure 6: Converting String Dates to Datetime Format

Figure 6 shows the steps involved in standardising string-based dates using pd.to-datetime(). Both the admission date column and the discharge date column are converted to DateTime objects to ensure the reliability of all time references across time-divisible entities in line with further analysis and plotting techniques. By using errors='coerce', any poorly formatted entries are handled by replacing them with NaT, which keeps the set clear and eliminates broken pipelines in the future.

```
# Calculate duration of admission in days
df_cleaned['total_admission_days'] = (
    df_cleaned['date_discharge_full_date_desc'] - df_cleaned['date_admission_full_date_desc']
).dt.days

# Show first 10 rows of the relevant columns
print(df_cleaned[['date_admission_full_date_desc', 'date_discharge_full_date_desc', 'total_admission_days']].head(10))
```

```
  date_admission_full_date_desc date_discharge_full_date_desc  \
0                    2019-08-02                    2019-08-04
1                    2019-06-03                    2019-06-04
2                    2019-07-03                    2019-07-06
3                    2019-09-03                    2019-09-11
4                    2019-11-04                    2019-11-11
5                    2019-01-05                    2019-01-10
6                    2019-02-05                    2019-02-15
7                    2019-07-05                    2019-07-07
8                    2019-01-06                    2019-01-09
9                    2019-06-06                    2019-06-07

   total_admission_days
0                   2.0
1                   1.0
2                   3.0
3                   8.0
4                   7.0
5                   5.0
6                  10.0
7                   2.0
8                   3.0
9                   1.0
```

Figure 7: Calculating Total Admission Duration in Days

Figure 7 above shows how the total admission period per patient in days is established. Here, the discharge date is simply subtracted from the date of admission, and the difference is recorded in a new column named total_admission_days. To test the precision, the first 10 records of the corresponding columns are printed, providing a rapid test and providing an idea of the variety of the lengths of the hospital stays among the population of the test.

```
import pandas as pd

# Load the cleaned CSV file back into memory
df_cleaned = pd.read_csv("Cleaned_Digestive_Disorder_2019_2019.csv")

# Confirm successful load
print(f"Data loaded. Shape: {df_cleaned.shape}")
```

```
Data loaded. Shape: (149750, 15)
```

Figure 8: Verifying Data Integrity by Reloading the Cleaned Dataset

Figure 8 above demonstrates how to reload the cleaned data by using a CSV file and then verify that the loading took place. Printing the shape of the data frame (k rows and columns) gives a speedy check against forgotten saves or data loss during the cleaning process. This test ensures that all observations

and characteristics are recorded, and it serves as preliminary quality assurance prior to any further analysis. By saving the clean version of the set to disk and importing it back, researchers can be certain that they deal with secure, uncorrupted values in further calculations and graphs. Shape verification is used to immediately identify a mismatch and to confirm that all cleaning operations have been completed.

3 Results and Discussion

GutVision facilitates the project requirements, as it will present national information about digestive disorders in Malaysia, and this is critical to the project needs. The same value was given to the front-end and the back-end parts of the system so that the system is usable and reliable. A clear explanation of GutVision is presented in data processing (back-end) and dashboard visualisation (front-end), which can aid the evaluation of prevalence and trends of digestive disorders amongst demographics and regions.



Figure 9: GutVision Dashboard – Main Page

Figure 9 shows that the GutVision main page has the fundamental elements of navigation that enable users to go across the various sections of the dashboard with ease. A navigation button links the main page and the other sections; there is also a Back to Home button that takes the user to the home page for convenience. This page indicates two prominent chart characteristics. The top part is a clickable table of the most widespread digestive diseases with a brief description under each tile to help individuals who may not recognise medical terminology. The second part is a state-level map that displays the total number of admissions due to digestive disorders in Malaysia between 2019 and 2023, which can be viewed by hovering over the dots of the map to display the number of annual admissions using the hover features.

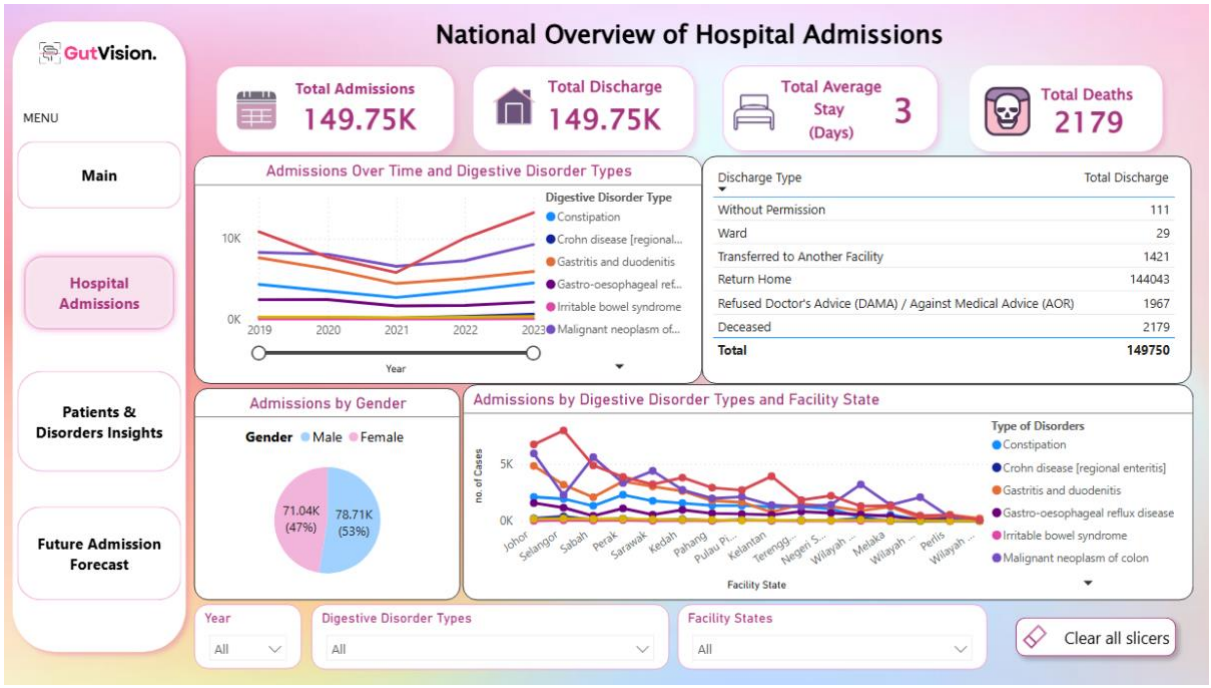

Figure 10: GutVision Dashboard – National Overview of Hospital Admissions Page

Figure 10 shows the National Overview of Hospital Admissions page, where users can filter this data by further breaking it down by year, facility state, and digestive disorder type to narrow the data shown. Key indicators are presented in card visuals according to total admissions, total discharges, average length of stay, and total deaths in 2019-2023. Navigation buttons allow a smooth change of pages, and the Back and Continue buttons allow one to read more information. The visuals on this page include a graph on admissions over the years and the types of disorders, which are helpful to see which year has the most or fewest hospitalisations. To provide a better understanding of admission effects, a discharge table shows home return, transfer to another facility, home discharge against medical advice (DAMA), or even death. A gender-based pie chart shows an analysis of the difference between males' and females' admissions, and the line chart, which combines the types of digestive disorders and facility states, reveals the most common conditions as well as the states with the highest admissions, with Johor, Selangor, and Sabah on top, consistent with their status as among Malaysia's most populous states, though further analysis would be needed to determine exact causal factors. Male patients accounted for 53% of admissions compared to females with 47%, suggesting potential gender-based patterns in digestive disorder prevalence or healthcare-seeking behaviour.

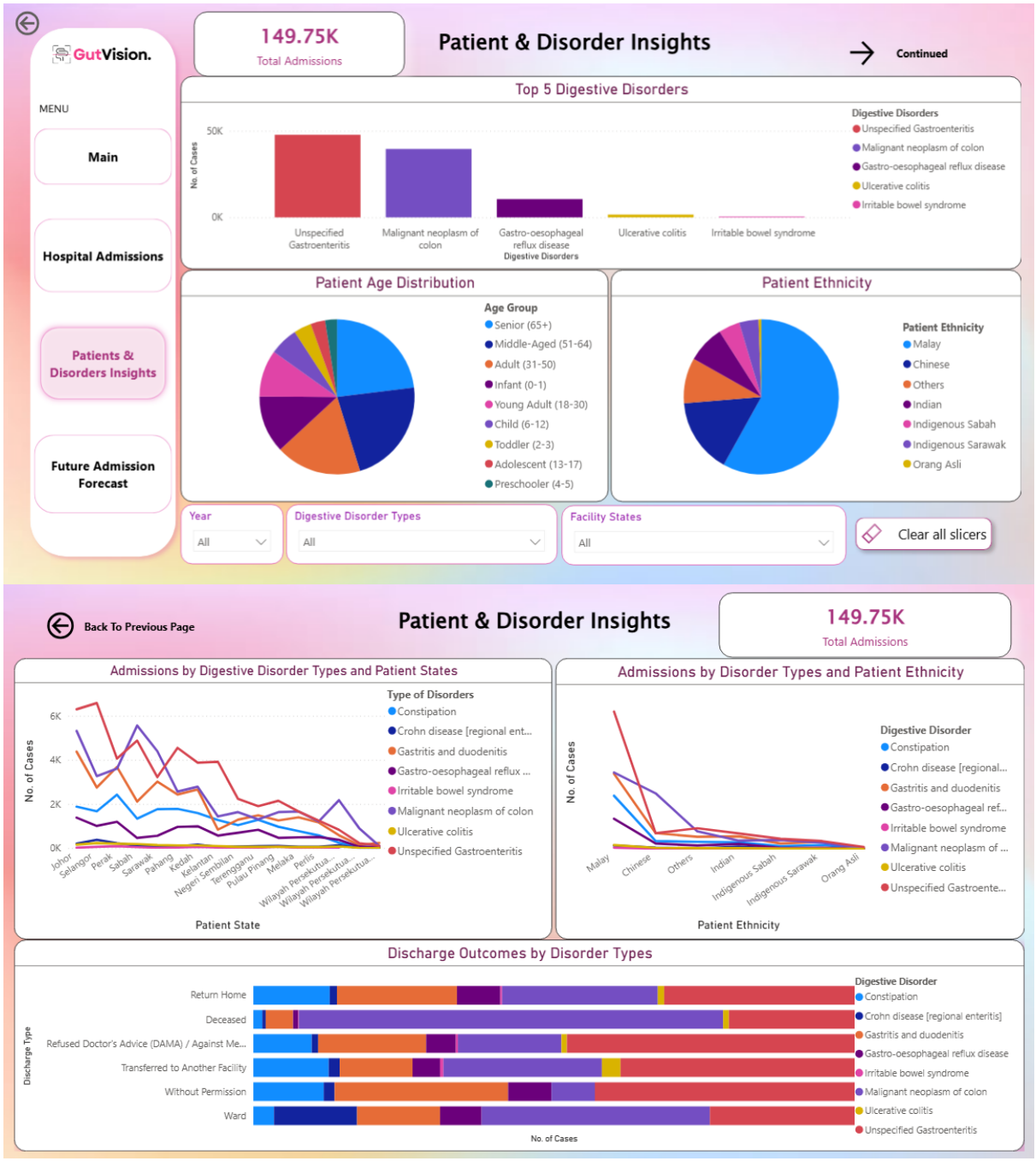

Figure 11: GutVision Dashboard – Patient and Disorder Insights Page

Figure 11 shows the Patient and Disorder Insights pages offer three filters: year, facility state, and disorder type that will allow narrowing the data to specific groups. There is a navigation button to help move between sections of the dashboard, and Back and Continue buttons to enhance accessibility. The key charts are bar charts of the top five digestive disorders and an age distribution that shows that senior patients, with 23.08%, and the middle-aged groups, 22.13%, register the greatest admissions, demonstrating a clear age-related pattern in digestive disorder prevalence. Estimates of the ethnicity indicate that the largest numbers of patients admitted are the Malays, followed by the Chinese. Malay ethnic groups show the highest rates at 58.06%, reflecting Malaysia's demographic distribution. Additional charts are linked to types of disorders with states, such that Johor, Selangor, and Perak are the most involved states; and to disorder types with ethnicity, such that Malays, with the highest number of unspecified gastroenteritis with 31,375 cases, and Chinese, with malignant neoplasm of the colon, with 12,681, are the most affected ethnic groups. Another chart examines the discharge outcomes; the highest mortality is recorded for the malignant neoplasm of the colon at 70.49%, and the highest number

of transfers occurs for unspecified gastroenteritis at 38.99%, which indicates a need for special care. Such insights inform resource planning and prioritise the most-at-risk groups of the population.

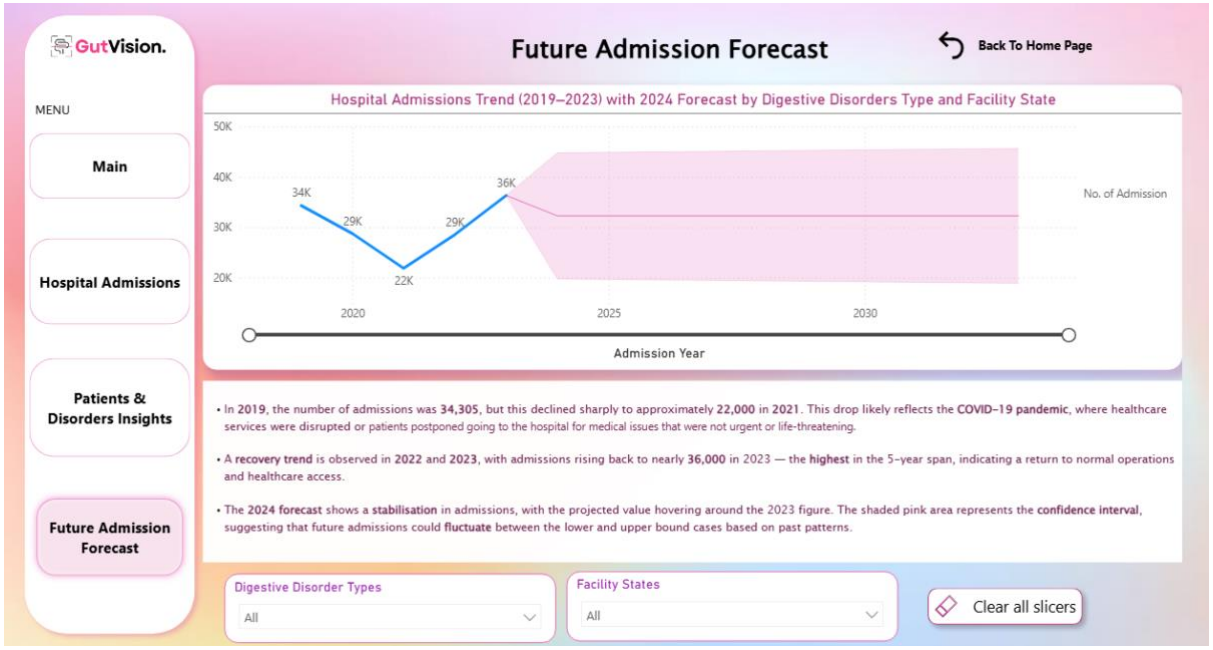

Figure 12: GutVision Dashboard - Future Admission Forecast Page

Figure 12 shows the Future Admission Forecast page using Power BI's exponential smoothing algorithm. While the model provides valuable trend projections, its accuracy is inherently limited by the COVID-19 pandemic disruptions visible in 2020-2021 admission data. The confidence bands indicate forecast uncertainty, reminding users that projections represent probable ranges rather than absolute predictions. This chart can be filtered by disorder and facility state so that users can forecast future admissions by condition or region. This can assist hospitals in strategic planning by matching staff, resources, and space to predicted demand, underspending, and optimising patient flow. A navigation button is used to move easily around the dashboard and a Back to Home button to navigate back to the home page. The central chart depicts trends in historical admissions (2019-2023) and a forecast for 2024, which enables its strong adopter to keenly identify any upward or downward changes in demand. The insights section considers the figures in the context that during COVID-19, there was a decline in admissions, though accuracy is limited by pandemic disruptions in 2020-2021, but in 2022 and 2023, the situation improved, and projections expect a further increase. Confidence bands are also present to cover the forecast uncertainty so that users understand results as ranges, but not as absolute forecasts.

4 Conclusion

The GutVision initiative is a significant contribution to tracking the trends in digestive health in Malaysia and accomplishes its task by utilising official data, technological advancements, and interactive visualisations. The project first established a strong basis of analysis using records in Pusat Informatik Kesihatan (2019-2023), complete with age, gender, region, and disorders including IBS and IBD. A Power BI dashboard was created, and this data was displayed in an interactive and intuitive manner where we were able to filter by year, state, age, and gender. The tooltips, sliders, buttons, and other features allow users to browse the patterns and see insights, and Agile sprints and the OSEMN cycle allow iterative improvement based on comments. The use of GutVision not only can help healthcare providers and the population understand the current trends and make informed choices but also make resource allocation and policy creation more effective. However, one limitation of GutVision

is: it uses historical batch-processed data rather than real-time streams and reliance on descriptive analytics without machine learning models. Future enhancements could include real-time data integration, AI/ML models for predictive analytics, and mobile compatibility for broader accessibility. In summary, GutVision is an excellent example of how data and technology can be used to solve health issues, and it could continue to grow in the future by incorporating other data and new features to aid in monitoring, education, and healthier decision-making in the Malaysian context.

Acknowledgements

The authors thank Pusat Informatik Kesihatan, Ministry of Health Malaysia for the approval to use official health data in this project.

Conflict of Interest Statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author Contribution

Norisan Abd Karim and Siti Aishah Mohmad Saaid contributed to the literature review, system design and development and preparation of the manuscript. Fauziah Redzuan and Fariza Hanis Abdul Razak contributed to the research content, interpretation of the findings, and review of the manuscript. Rogayah Abdul Majib and Nor Diana Ahmad reviewed the research content and the final manuscript before submission.

References

[1] S. Y. Lim, V. Ulaganathan, P. Nallamuthu, B. Gunasekaran, and S. Salvamani, "Dietary patterns and lifestyle factors associated with the risk of colorectal cancer: A hospital-based case-control study among Malaysians," *Malaysian Journal of Medical Sciences*, vol. 31, no. 1, pp. 212–234, Feb. 2024. doi: 10.21315/mjms2024.31.1.18.

[2] Ministry of Health Malaysia, "Data catalogue," *KKMNOW*. [Online]. Available: https://data.moh.gov.my/data-catalogue. Accessed: Nov. 30, 2025.

[3] X. T. Goh, S. K. Fong, H. C. Chai, B. P. Kee, and K. H. Chua, "The first association study of protein tyrosine phosphatase, non-receptor type 2 (PTPN2) gene polymorphisms in Malaysian patients with Crohn's disease," *Gene*, vol. 836, Art. no. 146661, Aug. 2022. doi: 10.1016/j.gene.2022.146661.

[4] T. P. T. Armand, K. A. Nfor, J.-I. Kim, and H.-C. Kim, "Applications of artificial intelligence, machine learning, and deep learning in nutrition: A systematic review," *Nutrients*, vol. 16, no. 7, Art. no. 1073, Apr. 2024. doi: 10.3390/nu16071073.

[5] Z. Ma, B. N. Jørgensen, and Z. G. Ma, "A systematic data characteristic understanding framework towards physical-sensor big data challenges," *Journal of Big Data*, vol. 11, Art. no. 84, Jun. 2024. doi: 10.1186/s40537-024-00942-5.

[6] K.-H. Yoo, C. K. Leung, and A. Nasridinov, "Big data analysis and visualization: Challenges and solutions," *Applied Sciences*, vol. 12, no. 16, Art. no. 8248, Aug. 2022. doi: 10.3390/app12168248.

[7] S. Kim, Y. Zhang, K. Mauldin, and B. N. Limketkai, "Advances in digital health for gastrointestinal nutrition," *Current Treatment Options in Gastroenterology*, vol. 22, pp. 267–278, 2024. doi: 10.1007/s11938-024-00458-0.

[8] R. Rabiei, P. Bastani, H. Ahmadi, S. Dehghan, and S. Almasi, "Developing public health surveillance dashboards: A scoping review on the design principles," *BMC Public Health*, vol. 24, Art. no. 392, Feb. 2024. doi: 10.1186/s12889-024-17841-2.

[9] J. Zhu *et al*., "Statistical modeling of gut microbiota for personalized health status monitoring," *Microbiome*, vol. 11, Art. no. 184, Aug. 2023. doi: 10.1186/s40168-023-01614-x.

[10] A. Jamarani, S. Haddadi, R. Sarvizadeh, M. H. Kashani, M. Akbari, and S. Moradi, "Big data and predictive analytics: A systematic review of applications," *Artificial Intelligence Review*, vol. 57, Art. no. 176, 2024. doi: 10.1007/s10462-024-10811-5.

[11] M. Malkani, E. Madan, D. Malkani, A. Madan, N. Singh, T. Bamji, and H. Sabharwal, "Rank ordered design attributes for health care dashboards including artificial intelligence: Usability study," *Online Journal of Public Health Informatics*, vol. 16, Art. no. e58277, Nov. 2024. doi: 10.2196/58277.

[12] University of Minnesota, "Agile methodology: Advantages and disadvantages," *University of Minnesota*, Feb. 11, 2022. [Online]. Available: https://ccaps.umn.edu/story/agile-methodology-advantages-and-disadvantages. Accessed: Nov. 17, 2026.

[13] K. Kumari, M. Bhardwaj, and S. Sharma, "OSEMN approach for real time data analysis," *International Journal of Engineering and Management Research*, vol. 10, no. 2, pp. 107–110, Apr. 2020. doi: 10.31033/ijemr.10.2.11.

[14] M. Asgari and T. Hurtut, "A design language for prototyping and storyboarding data-driven stories," Applied Sciences, vol. 14, no. 4, Art. no. 1387, Feb. 2024, doi: 10.3390/app14041387

[15] M. Huisman, "The 5 steps to data visualization," *LinkedIn*, Sep. 6, 2024. [Online]. Available: https://www.linkedin.com/pulse/5-steps-data-visualization-menno-huisman-0qa9e/. Accessed: Nov. 17, 2026.

[16] UXPin, "Dashboard Design Principles: The Definitive Guide (2026)" Studio by UXPin, June. 24, 2026. [Online]. Available: https://www.uxpin.com/studio/blog/dashboard-design-principles/. Accessed: Nov. 17, 2026.