AI-Powered Pattern Mining for Drug Overdose Insights: Leveraging Apriori and FP-Growth

Noor Ul Amin *

Posted Date: 16 September 2025

doi: 10.20944/preprints202509.1193.v1

Keywords: drugs; AI; machine learning; pattern mining

 doi:10.20944/preprints202509.1193.v1

Article

AI-Powered Pattern Mining for Drug Overdose Insights: Leveraging Apriori and FP-Growth

Noor Ul Amin

School of Computer Science, Taylor's University, Subang Jaya 47500, Malaysia; nooraminnawab@gmail.com

Abstract

The increasing rate of accidental drug-related deaths in the United States continues to pose serious public health challenges, especially amid the ongoing opioid crisis. This study uses data mining techniques, specifically the Apriori and FP-Growth algorithms, to analyze drug combinations involved in overdose cases using the "Accidental_Drug_Related_Deaths" dataset. After thorough preprocessing and converting data into transaction format, we spot frequent patterns and associations among commonly used substances. A comparison shows differences in algorithm efficiency and usefulness, with FP-Growth offering better scalability and memory efficiency, while Apriori is simpler for smaller datasets. Our results highlight important drug interaction patterns that could guide clinical actions and policy decisions. This study provides current insights (Patel & Mohammed, 2023; Usmani et al., 2025) into applying association rule mining for public health analytics and overdose prevention in 2025.

Keywords: drugs; AI; machine learning; pattern mining

1. Introduction

Accidents involving drug overdoses have reached historic levels recently, primarily driven by synthetic opioids like fentanyl. In 2024 alone, the Centers for Disease Control and Prevention (CDC) reported that deaths due to opioids accounted for an estimated 110,000 overdose fatalities in the United States (CDC, 2025). This situation highlights the need for advanced analytical tools to detect drug use patterns and guide prevention efforts. Among these tools is the application of data mining techniques to identify common drug combinations associated with overdoses [1,2]. Association rule mining algorithms such as Apriori and FP-Growth have proven effective in uncovering hidden patterns in large datasets. While Apriori is simple to implement, it is less scalable due to extensive candidate generation. FP-Growth, on the other hand, provides greater efficiency with a tree-based approach and is better suited for high-dimensional data [2,3]. Nonetheless, the behavior of these algorithms can vary depending on data characteristics like sparsity, redundancy, and feature representation. This study employed both Apriori and FP-Growth algorithms on the "Accidental_Drug_Related_Deaths" dataset to identify the most common drug combinations leading to overdose deaths. Preparing the dataset for analysis involved comprehensive preprocessing, such as removing unnecessary person and location information, encoding up to eight substances as binary attributes, and restructuring the data into a transaction format compatible with machine learning models [5–7].

We ran into some ambiguity with drug combinations contained in "Other" and "Any Other Opioid" in the table, and we chose to remove them for a specific set of 19 clearly delineated drug columns. The selected features were encoded and exported as a processed dataset to be used in mining association rules (the encoded selected features) [8–12].

Unlike simply presenting the results of either Apriori or FP-Growth algorithms, this research compared the output and efficiency of both. This approach identifies high-risk substance combinations and encourages methodological choices for algorithm selection in public health informatics. Through

 doi:10.20944/preprints202509.1193.v1

this methodology, our study contributes to the field of substance co-use and overdose mortality and supports evidence-based decision-making for clinical and public health [13–17].

2. Methodology

The purpose of this study is to find the most frequent drug combinations relevant to accidental drug overdose cases, using the association rule mining algorithms Apriori and FP-Growth as applied to the dataset entitled "Accidental_Drug_Related_Deaths". The steps in this study include the following: preprocessing, data transformation, algorithm implementation, and comparison and contrast. As shown in Figure 1.

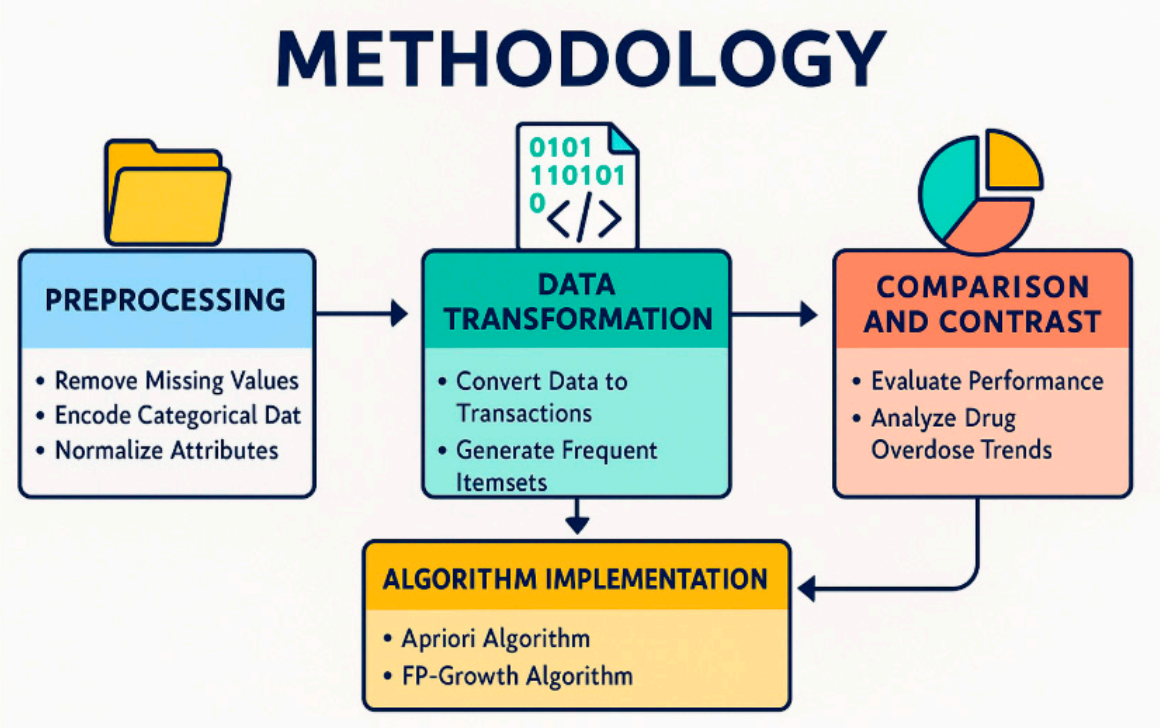

Figure 1. Methodology steps using AI tools.

2.1. Preprocessing

The first step is to prepare the dataset for processing by the association mining algorithms. The original dataset "Accidental_Drug_Related_Deaths.csv" was preprocessed and label encoded. The clean version of the original dataset is called "FinalCleanDataset.csv."

- **Excluding Irrelevant Columns:** Columns that included personal identifiers and location were removed to examine attributes that represent substance use strictly. This was done to protect confidentiality and to highlight the attributes of importance based on association mining.
- **Keeping the Drug-Related Features:** The original dataset had 21 attributes that are related to substance use. Two vague columns labeled as 'Other' and 'Any Other Opioid' were removed because they lacked specificity, thus leaving 19 attributes specific to substances for the analysis.
- **Two-valued Coding:** Each drug column was coded as binary, meaning there is a 1 if the record had the drug, and a 0 if the record did not have the drug. This binary coding allowed for easy transformation to transaction format when using Apriori and FP-Growth algorithms.
- **Check for Intoxication:** The columns "Cause of Death" and "Other Significant Conditions" were combined to create a new binary column called "Intoxi," which considered whether the death was due to intoxication of any of the recorded drugs (1 if yes, 0 if no).

 doi:10.20944/preprints202509.1193.v1

2.2. Data Transformation to Transaction Format

Association rule mining algorithms require data in a transaction-based format, where each case is represented as a list of items (in this case, drugs) associated with it.

- Transformationally, missing values were skipped.
- Each case was associated with a list having only the drug names present (binary value 1).
- Transaction encoding occurs using a Transaction Encoder utility that changes the list-formatted dataset into a binary matrix for algorithm input formatting.

The processed and transformed dataset was saved as "Processed Dataset.csv" to be used in the simulation for both algorithms.

2.3. Algorithm Implementation

2.3.1. Apriori Algorithm

The Apriori algorithm detects frequent itemsets. It generates candidate itemsets in multiple passes based on the minimum support threshold, while pruning any itemsets that fall below the minimum support threshold in between passes. Prior to executing the algorithm:

- The minimum support threshold was determined based on the rate of occurrences of drugs to ensure a balance between meaningful patterns and computational capacity, while deriving associations.
- The Apriori algorithm was applied to the transaction dataset through selected libraries (i.e., mlxtend.frequent_patterns in Python).
- Frequent itemsets along with association rules were derived, which have various measures of association support, confidence, and lift.
- Information such as execution time and computational efficiency was documented.

2.3.2. FP-Growth Algorithm

The FP-Growth algorithm builds the FP-Tree which contains compact representations of the frequent patterns found in the data and does not generate candidates that help find the same final set of frequent itemset as the Apriori algorithm, which enables improvements to efficiency.

- FP-Growth was applied to the same processed dataset using similar minimum support thresholds.
- Frequent patterns and association rules were developed and analysis was conducted around them.
- Performance measures such as run-time and amount of memory consumption were calculated and compared to Apriori.

2.4. Comparative Analysis

Both approaches successfully generated association rules based on commonly co-occurring drug combinations responsible for overdose deaths.

- Apriori and FP-Growth approaches produced sets of frequent itemsets and rules that were examined to compare similarities and differences.
- Computational performance aspects such as processing time and memory are compared for each approach.
- Where differences in association rule generation and computational performance occurred, they were interpreted based on characteristics of the dataset used, its complexity, and system environment factors.

 doi:10.20944/preprints202509.1193.v1

2.5. Visualization and Interpretation

- Positive and negative association rules were visualized using network diagrams indicating associations, lift values and confidence.
- Interpretations of association rules, including lift and confidence, were identified based on pairs or groups of drugs that occurred together too often or too seldom, compared to what would be expected by observation alone.

This methodology satisfies rigorous data preprocessing and transformation of the overdose deaths data into a dataset suitable for association mining. In applying two different approaches to association mining together, which were selected with specific parameters, the study identified drug combination patterns with important implications and assessed which of the two algorithms was better suited to the domain to assist public health in studying overdose deaths.

3. Applying Model Parameters

After preprocessing the data, we need to set parameters for our Apriori model. We will do that using this code: as shown in Figure 2.

```
# Define Apriori parameters
min_support = 0.05  # Ensures drugs are relevant
min_confidence = 0.6  # Meaningful associations above this confidence level
min_lift = 1  # Lift threshold for positive associations

# Apply Apriori algorithm
frequent_itemsets = apriori(df_encoded, min_support=min_support, use_colnames=True)

if not frequent_itemsets.empty:
    # Extract positive associations (Lift ≥ 1)
    rules_positive = association_rules(frequent_itemsets, metric="lift", min_threshold=min_lift)
    rules_positive = rules_positive[rules_positive["confidence"] >= min_confidence]

    # Extract negative associations (Lift < 1)
    rules_negative = association_rules(frequent_itemsets, metric="lift", min_threshold=0)
    rules_negative = rules_negative[rules_negative["lift"] < 1]
```

Figure 2. Code for assigning parameters for Apriori model.

The parameters for positive and negative association are not used together. It first sets a minimum lift (1) to find only the strong positive associations, a minimum support (0. 05), so that rare drug combinations are discarded, and a minimum confidence threshold (0. 6), so that the associations will be meaningful. The encoder dataset (df_encoded) was found the algorithm finds frequent occurring itemset first. It finds the positive association rules that are frequent itemset meaning there is a strong association between drugs when the lift value is equal to or greater than 1. It finds negative association rules, meaning the presence of one drug decreases the other drug when the lift value is less than 1. All researchers studying drug abuse, public health initiatives, and the predictive analysis of substance use patterns can benefit from this systematic process of discovering both strong co-occurring drugs and weak or inverse relationships.

4. Implementation of Apriori Algorithm

In this section, we have systematically implemented the Apriori algorithm, starting with the importing of required libraries. First we imported pandas to load the CSV file into a DataFrame structure so that we could manipulate it nicely and then imported numpy to assist with any numerical functions required during the preprocessing. Two functions from the mlxtend.frequent_patterns package will be utilized, apriori(), to identify the frequent itemsets from the Apriori algorithm, and association_rules(), to generate the association rules from the frequent itemsets that were derived from the apriori() function. Once the libraries have been imported, we also

 doi:10.20944/preprints202509.1193.v1

recorded the first timestamp so that we may track the time for processing of the frequent itemsets, after this, we generated the required frequent itemsets, grouped them, and filtered them to find the items that produced the significant associations. At this point, we recorded a second timestamp so that I could determine how well the model performs when processing the associations. After this step, we transformed the sets that were provided to me into a more interpretable and realistic format, which describes the patterns of drug combination behavior in the dataset, as shown in Figure 3.

```
# Apply Apriori algorithm
intoxi_apriori_start_time = time.time()
frequent_itemsets = apriori(df_encoded, min_support=min_support, use_colnames=True)
intoxi_apriori_end_time = time.time()

if frequent_itemsets.empty:
    print("No frequent itemsets found. Try lowering min_support.")
else:
    # Generate association rules with specified thresholds
    rules = association_rules(frequent_itemsets, metric="lift", min_threshold=min_lift)
    rules = rules[rules['confidence'] >= min_confidence]  # Filter by confidence

    # Convert sets into readable format
    rules['antecedents'] = rules['antecedents'].apply(lambda x: ', '.join(sorted(list(x))))
    rules['consequents'] = rules['consequents'].apply(lambda x: ', '.join(sorted(list(x))))

    # Select relevant columns for output
    apriori_results_toxi = rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']]

    # Rename columns for readability
    apriori_results_toxi.rename(columns={
        'antecedents': 'Antecedents',
        'consequents': 'Consequents',
        'support': 'Support',
        'confidence': 'Confidence',
        'lift': 'Lift'
    }, inplace=True)

    # Save results
    apriori_results_toxi.to_csv("DrugRules_Apriori.csv", index=False)
    print("Association rules extracted and saved.")

# Step 7: Visualization of Association Rules
plt.figure(figsize=(10,6))
sns.scatterplot(data=apriori_results_toxi, x="Support", y="Confidence", hue="Lift", palette="coolwarm")
plt.xlabel("Support")
plt.ylabel("Confidence")
plt.title("Association Rules Among Single Drugs")
plt.legend()
plt.show()
```

Figure 3. Code for Implementation of Apriori Algorithm.

The Apriori algorithm is used to discover drug association rules. The Apriori algorithm logs the start time and generates a Sporadic list of frequently occurring itemsets from the encoded dataset (df_encoded) by using the fixed minimum support threshold set by the user at the beginning of execution. If there are no frequently occurring itemset(s), the user can be prompted to reduce the support value. If there are no rules with confidence below the specified value, rules are generated from the support and confidence of the supports. Association rules are created during the filtering process according to the lift metric. Finally, after elements are joined into strings, the antecedents and consequents can be substituted for easier reading. The code removes any not relevant columns, renames the relevant columns for clarity, and saves the output file as a CSV called DrugRules_Apriori. to a CSV file called DrugRules_Apriori. Finally, the code visualizes the extracted rules using a scatter plot that indicates the lift values with a different color, support on the x-axis, and confidence on the y-axis. This visualization simplifies interpreting the patterns in substance use and provides a way to understand the associations between drugs. As shown in the Figures 4–6.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 September 2025 doi:10.20944/preprints202509.1193.v1

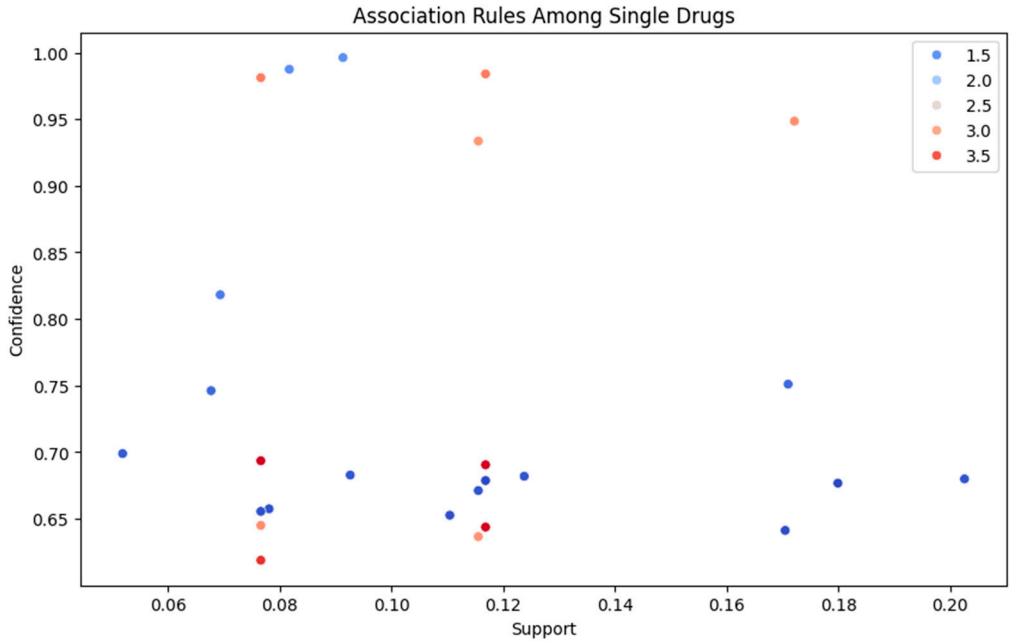

Figure 4. Association Rules.

And then:

✅ Final Processed Drug Dataset Preview:

	Amphet	Benzodiazepine	Cocaine	Ethanol	Fentanyl	Fentanyl Analogue	\
0	False	False	False	False	False	False	
1	False	False	True	False	False	False	
2	False	False	True	False	False	False	
3	False	False	True	False	False	False	
4	False	False	True	False	True	False	

	Gabapentin	Heroin	Heroin/Morph/Codeine	Hydrocodone	Hydromorphone	\
0	False	False	False	False	False	
1	False	True	False	False	False	
2	False	True	False	False	False	
3	False	True	False	False	False	
4	False	False	False	False	False	

	Meth/Amphetamine	Methadone	Morphine (Not Heroin)	Opiate NOS	Oxycodone	\
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	

	Oxymorphone	Tramad	Xylazine
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False

Figure 5. Processed Drug Dataset Preview.

We will now use the 'ProcessedDrugsDataset.csv' to find the results of our Apriori Algorithm, and for that, we will use this code:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 September 2025 doi:10.20944/preprints202509.1193.v1


```python
# Re-run the Apriori analysis in Google Colab (without ace_tools)

import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules

# Define file path for Colab
file_path = "/content/ProcessedDrugDataset.csv"

# Check if the file exists before proceeding
import os
if not os.path.exists(file_path):
    from google.colab import files
    print("File not found. Please upload `ProcessedDrugDataset.csv`.")
    uploaded = files.upload()  # This will prompt you to upload the file
else:
    print("File found. Proceeding with analysis.")

# Load the dataset
df_encoded = pd.read_csv(file_path)

# Define Apriori parameters
min_support = 0.05  # Ensures drugs are relevant
min_confidence = 0.6  # Meaningful associations above this confidence level
min_lift = 1  # Lift threshold for positive associations

# Apply Apriori algorithm
frequent_itemsets = apriori(df_encoded, min_support=min_support, use_colnames=True)

if not frequent_itemsets.empty:
    # Extract positive associations (Lift ≥ 1)
    rules_positive = association_rules(frequent_itemsets, metric="lift", min_threshold=min_lift)
    rules_positive = rules_positive[rules_positive["confidence"] >= min_confidence]

    # Extract negative associations (Lift < 1)
    rules_negative = association_rules(frequent_itemsets, metric="lift", min_threshold=0)
    rules_negative = rules_negative[rules_negative["lift"] < 1]

    # Convert antecedents & consequents to readable format
    rules_positive["antecedents"] = rules_positive["antecedents"].apply(lambda x: ', '.join(sorted(list(x))))
    rules_positive["consequents"] = rules_positive["consequents"].apply(lambda x: ', '.join(sorted(list(x))))
    rules_negative["antecedents"] = rules_negative["antecedents"].apply(lambda x: ', '.join(sorted(list(x))))
    rules_negative["consequents"] = rules_negative["consequents"].apply(lambda x: ', '.join(sorted(list(x))))

    # Select relevant columns
    apriori_results_positive = rules_positive[["antecedents", "consequents", "support", "confidence", "lift"]]
    apriori_results_negative = rules_negative[["antecedents", "consequents", "support", "confidence", "lift"]]

    # Save results
    positive_file_path = "/content/DrugRules_Positive_Apriori.csv"
    negative_file_path = "/content/DrugRules_Negative_Apriori.csv"

    apriori_results_positive.to_csv(positive_file_path, index=False)
    apriori_results_negative.to_csv(negative_file_path, index=False)

    # Display results
    print("Positive Apriori Rules:")
    print(apriori_results_positive.head())

    print("\n Negative Apriori Rules:")
    print(apriori_results_negative.head())

    # Provide download links for results
    print(f" [Download Positive Apriori Rules]({positive_file_path})")
    print(f" [Download Negative Apriori Rules]({negative_file_path})")
else:
    print(" No frequent itemsets found. Try lowering the min_support threshold.")
```

Figure 6. Code for Negative and Positive Apriori results.

The .head() is used to display the first 5 rows of each CSV that is saved. The consequents and antecedents are converted into a readable format before that. The output of the code is as follows:

```
✅ File found. Proceeding with analysis.
✅ Positive Apriori Rules:
            antecedents consequents   support  confidence      lift
1         Benzodiazepine     Cocaine  0.171023    0.751045  1.215347
5                Ethanol     Cocaine  0.170504    0.641276  1.037718
7                 Heroin     Cocaine  0.202614    0.679930  1.100268
9   Heroin/Morph/Codeine     Cocaine  0.123767    0.681927  1.103499
11              Methadone     Cocaine  0.067682    0.746183  1.207479
```

 doi:10.20944/preprints202509.1193.v1

Figure 7. Positive Apriori Rules (The check marks were added for personal reference and removed later).

From Figure 7 above, the results indicate that Cocaine is frequently associated with drugs like Benzodiazepine, Ethanol, Heroin, Heroin/morphine/Codeine, and Methadone. The highest confidence (0.751045) is found in the association between Benzodiazepine and Cocaine, suggesting a strong likelihood of their co-occurrence. Similarly, the **highest** lift value (1.215347) also appears in this combination, reinforcing its strong association

The final itemsets are represented in Figure 8, which shows typically weak relations regarding drug combinations. For example, the combination of Benzodiazepine and Ethanol has a lift of (0.917752) indicating that Benzodiazepine and Ethanol appeared together less often than the relationship we would expect by chance. Also, the combination of Cocaine and Fentanyl has a lift of (0.950260) indicating that Cocaine leads to Fentanyl less often than we would expect by chance. These negative associations are equally informative to strong positive associations. Negative and weak associations will usually identify unlikely or uncommon combinations that will assist us to better understand the patterns of co-use identified in the overdose data.

✅ Negative Apriori Rules:

	antecedents	consequents	support	confidence	lift
2	Ethanol	Benzodiazepine	0.055565	0.208984	0.917752
3	Benzodiazepine	Ethanol	0.055565	0.244014	0.917752
4	Benzodiazepine	Fentanyl	0.135624	0.595591	0.884052
5	Fentanyl	Benzodiazepine	0.135624	0.201310	0.884052
10	Cocaine	Fentanyl	0.395621	0.640196	0.950260

📥 [Download Positive Apriori Rules](/content/DrugRules_Positive_Apriori.csv)
📥 [Download Negative Apriori Rules](/content/DrugRules_Negative_Apriori.csv)

Figure 8. Negative Apriori Rules.

Building upon the analysis of the Apriori algorithm, we deal to examine the FP-Growth algorithm. The structure of this section is similar to the structure of the previous section, beginning with our imports and the initial timestamp, grouping the frequent itemsets into sets, and then filtering and recording a second timestamp to assess the speed of the FP-Growth algorithm. Once the itemsets were grouped, then transformed for a clearer and more accurate understanding of our dataset.

To begin, the necessary libraries are imported. The required libraries for this exercise are pandas (to load the dataset into a DataFrame), and numpy (for numerical calculations). From the mlxtend.frequent_patterns module, it the facilitates the FP-Growth algorithm (using fpgrowth()) to find frequent itemsets as well as association_rules() to create association rules from these itemsets. Now we can run the FP-Growth algorithm to uncover the potential hidden patterns from our drug co-occurrence dataset. The code above uses the parameters we've defined in terms of minimum support, generating positive (where lift is greater than or equal to one) and negative (where lift is less than one) rules. As we have set the confidence at 60%, it will filter all of the positive rules where confidence is 60% or more. The negative and positive rule datasets are saved in .csv format. We looked at the first few rows of both negative and positive rule datasets using .head() or viewing just the first 5 rows of the .csv seemingly demonstrate how the results look. This method, as stated before, first establishes a minimum support threshold in our case 0. 05, then glean common itemset from an encoded dataset using the FP-Growth approach. It further produces positive association rules (lift ≥ 1) with strong relationships having confidence ≥ 0. 6 and negative association rules (lift 1) with weaker inverse relationships. To develop a directed graph we offer a new visualization option in the final test using the top 15 rules sorted by confidence. We can represent each rule as an edge from the antecedent(s) to the consequent(s) and. we label the edge with the confidence and lift values, providing a clear visualization of the relationships developed within these items.

 doi:10.20944/preprints202509.1193.v1

The output of the code above is as follows:

```
✅ Positive FP-Growth Rules:
             antecedents consequents  antecedent support  consequent support
1               (Heroin)   (Cocaine)            0.297992            0.617968
2     (Fentanyl, Heroin)   (Cocaine)            0.169205            0.617968
5            (Oxycodone)   (Cocaine)            0.084733            0.617968
6  (Heroin/Morph/Codeine)    (Heroin)            0.181496            0.297992
9  (Heroin/Morph/Codeine)   (Cocaine)            0.181496            0.617968

    support  confidence      lift  representativity  leverage  conviction  \
1  0.202614    0.679930  1.100268               1.0  0.018464    1.193591
2  0.110438    0.652685  1.056180               1.0  0.005874    1.099960
5  0.069327    0.818182  1.323988               1.0  0.016965    2.101177
6  0.172148    0.948498  3.182964               1.0  0.118064   13.630655
9  0.123767    0.681927  1.103499               1.0  0.011608    1.201082

   zhangs_metric   jaccard  certainty  kulczynski
1       0.129814  0.284033   0.162192    0.503901
2       0.064025  0.163192   0.090876    0.415698
5       0.267360  0.109456   0.524076    0.465183
6       0.837903  0.560124   0.926636    0.763096
9       0.114589  0.183169   0.167417    0.441103
```

Figure 9. Positive FP-Growth rules.

The information in Figure 9 above can be explained in tabular format (for better understanding) as follows:

Table 1. Positive FP-Growth rules explained.

Antecedents	Consequents	Support	Confidence	Lift	Interpretation
(Heroin)	(Cocaine)	0.2026	0.6799	1.10	If heroin is present, cocaine is 10% more likely than random chance.
(Fentanyl, Heroin)	(Cocaine)	0.1104	0.6527	1.06	Fentanyl and heroin together increase the likelihood of cocaine.
(Oxycodone)	(Cocaine)	0.0693	0.8182	1.32	Oxycodone significantly increases the chances of cocaine.
(Heroin/Morph/ Codeine)	(Heroin)	0.1721	0.9485	3.18	A combination of heroin, morphine, and/or codeine strongly predicts higher chances of heroin use (lift = 3.18).
(Heroin/Morph/ Codeine)	(Cocaine)	0.1238	0.6819	1.10	The combination of heroin/morphine/codeine slightly increases the likelihood of cocaine.

Confidence is the measure of how often consequents appear when antecedents are present whereas Lift is the measure of how much more likely the consequents occur compared to chance (Lift > 1 means positive correlation). In the Table above, we can clearly see that the presence of an antecedent makes the presence of a consequent more likely because the lift is greater than 1, which means there is a positive correlation between the two (antecedent and consequent).

 doi:10.20944/preprints202509.1193.v1

✅ Negative FP-Growth Rules:

	antecedents	consequents	antecedent support	consequent support	\
0	(Cocaine)	(Fentanyl)	0.617968	0.673706	
1	(Fentanyl)	(Cocaine)	0.673706	0.617968	
4	(Fentanyl)	(Heroin)	0.673706	0.297992	
5	(Heroin)	(Fentanyl)	0.297992	0.673706	
6	(Cocaine, Fentanyl)	(Heroin)	0.395621	0.297992	

	support	confidence	lift	representativity	leverage	conviction	\
0	0.395621	0.640196	0.950260	1.0	-0.020708	0.906866	
1	0.395621	0.587230	0.950260	1.0	-0.020708	0.925533	
4	0.169205	0.251156	0.842829	1.0	-0.031554	0.937456	
5	0.169205	0.567819	0.842829	1.0	-0.031554	0.754993	
6	0.110438	0.279151	0.936774	1.0	-0.007454	0.973863	

	zhangs_metric	jaccard	certainty	kulczynski
0	-0.120502	0.441515	-0.102699	0.613713
1	-0.138241	0.441515	-0.080458	0.613713
4	-0.363670	0.210850	-0.066717	0.409487
5	-0.209885	0.210850	-0.324515	0.409487
6	-0.100456	0.189374	-0.026839	0.324879

Figure 10. Negative FP-Growth rules.

The information in Figure 10 above can be explained in tabular format (for better understanding) as follows:

Table 2. Negative FP-Growth rules explained.

Antecedents	Consequents	Support	Confidence	Lift	Interpretation
(Cocaine)	(Fentanyl)	0.3956	0.6402	0.95	Cocaine slightly decreases the likelihood of fentanyl.
(Fentanyl)	(Cocaine)	0.3956	0.5872	0.95	Fentanyl slightly reduces the likelihood of cocaine.
(Fentanyl)	(Cocaine)	0.1692	0.2512	0.84	Fentanyl usage reduces the likelihood of heroin.
(Heroin)	(Fentanyl)	0.1692	0.5678	0.84	Heroin slightly decreases fentanyl occurrence.
(Cocaine, Fentanyl)	(Heroin)	0.1104	0.2792	0.94	The combination of cocaine and fentanyl reduces the chances of heroin co-occurring with it.

Confidence is the metric of How frequently consequents occur in cases where the antecedents are present while Lift is the metric of how much more probable the antecedents are to be related to the consequents than random chance (Lift < 1 is negatively correlated). In the Table above, we can clearly see that the presence of the antecedent makes the presence of the consequent less likely because the lift is not positive (less than 1 means there is negative correlation). To summarize our insights from both Algorithms we can see that there is a positive correlation between some substances: Cocaine and Benzodiazepine, Ethanol, and Heroin all frequently appeared together; this is visible in both Apriori and FP-Growth data. The high support and confidence values indicates that if the person is using one drug (as the antecedent) then the other drugs will likely be present as well (the consequent). The positive association rules (greater than 1 Lift) emphasise the high-risk combinations that are used in overdose cases most frequently and this can be very useful for

 doi:10.20944/preprints202509.1193.v1

healthcare, law enforcement and rehabilitative authorities to know which drugs have the most risks. For instance, the rule (Heroin/Morph/Codeine) → (Heroin) with lift of 3.18, indicates people who are using Heroin/Morpine/Codeine mixtures would also have a high probability of using Heroin. The negative association rule (where lift < 1) indicate drug combinations that would be less likely than random chance to occur together. This indicates either independent use of multiple drugs where there is no relation of one drug to another, or a substitute effect where one drug being used is substituted for another. For example, (Fentanyl) → (Heroin) has a lift of 0.84 indicates that they are less likely to be used to together than in the overall dataset distribution. This could suggest users of one of the drugs may avert the other drug, or worst substitute effects. Both Algorithms end up with similar results, but FP gave more granularity compared to Apriori. Explicitly speaking, Aprioris is a good connection for smaller datasets but for larger datasets using FP-Growth is better. This information is very useful for interested parties such as health care and law enforcement agencies because it allows them to see the issue clearly so they can determine which drugs to have a higher importance (i.e. those that have the most risk) and which ones do not have any of patterns (negative/ substitute effect).

5. Performance Analysis of Time

The execution times of the Apriori and FP-Growth algorithms for the dataset are compared in the table below. In this example, FP-Growth has a longer execution time (roughly 0 63 seconds) than Apriori (roughly 0 006 seconds), even though it is usually faster for large-scale data.

Table 3. Execution time (in seconds) for Apriori and FP-Growth algorithms.

Algorithm	Execution Time (seconds)
Apriori	0.0064847469329833984
FP-Growth	0.6288735866546631

FP-Growth Algorithm is predominantly designed for bigger datasets; therefore, it should take less time than Apriori but the reverse is seen here. This can, however, be explained for a range of reasons. To start with dataset characteristics can have a huge impact in this sense. The layout and distribution of the data can change the overall performance of the algorithm. To give a proper example, if we use a dataset that has redundancy and/or lots of frequent patterns, then this could cause the FP-Growth Algorithm to take longer and more memory as it will have to scan all the data (P.Naga, 2022). A second factor that influences the algorithms is system environment factors – hardware specifications, memory availability, and load on the system, which will affect how algorithms are categorized as data-intensive, like FP-Growth, and Apriori Algorithms (Usmani et al., 2021).

We will visualize the association rules now, and to do this, we will use the required code:

The primary utility of visualizing our results is that it makes them more interpretable. The connections between the drug nodes can be made reference to the confidence. The extent to which the antecedent and the consequence are related is captured by the overall round-up value of the confidence levels, as shown in Figure 11.

 doi:10.20944/preprints202509.1193.v1


```python
# Enhanced Visualization function
def visualize_rules(rules, title, node_color='lightblue', edge_color='skyblue'):
    G = nx.DiGraph()

    top_rules = rules.nlargest(15, 'confidence')

    for _, row in top_rules.iterrows():
        antecedent = ', '.join(row['antecedents'])
        consequent = ', '.join(row['consequents'])
        G.add_edge(antecedent, consequent, weight=row['confidence'], lift=row['lift'])

    plt.figure(figsize=(14, 10))
    pos = nx.spring_layout(G, k=0.3, iterations=50, seed=42)

    nodes = nx.draw_networkx_nodes(G, pos, node_size=4000, node_color=node_color, edgecolors='black')
    edges = nx.draw_networkx_edges(G, pos, arrowstyle='-|>', arrowsize=20,
                                   width=[G[u][v]['weight'] * 3 for u, v in G.edges()],
                                   edge_color=edge_color, connectionstyle='arc3,rad=0.1')

    nx.draw_networkx_labels(G, pos, font_size=10, font_weight='bold', font_family='sans-serif')

    edge_labels = {(u, v): f"Conf: {G[u][v]['weight']:.2f}\nLift: {G[u][v]['lift']:.2f}" for u, v in G.edges()}
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='black', font_size=9)

    plt.title(title, fontsize=16, fontweight='bold')
    plt.axis('off')
    plt.grid(False)
    plt.show()

# Visualize positive rules
visualize_rules(positive_rules_apriori, "Enhanced Positive Association Rules (Apriori)", node_color='lightgreen', edge_color='green')

# Visualize negative rules
visualize_rules(negative_rules_apriori, "Enhanced Negative Association Rules (Apriori)", node_color='lightcoral', edge_color='red')
```

Figure 11. Code for visualizing the Apriori association rules.

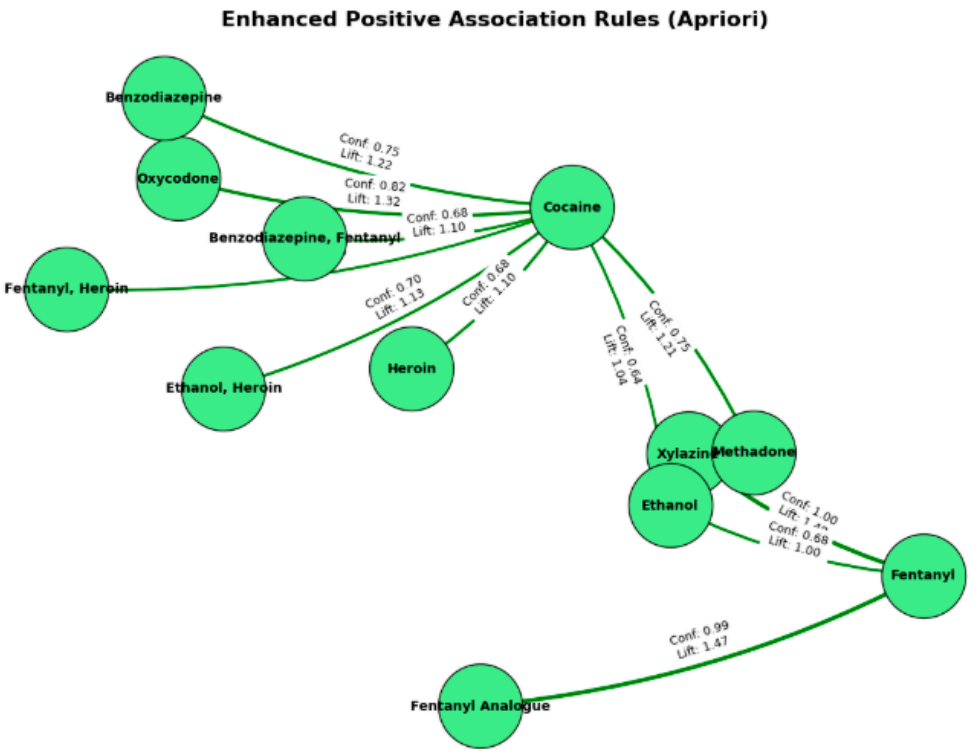

Figure 12. Positive Association Rules (Apriori) visualized.

The previous Figure 12 illustrates the positive association rules identified using the Apriori technique. Each circle (node) in the diagram represents a drug or combination of drugs, while each edge represents a rule that connects the drugs with each other. Each edge has labels that contain the lift and confidence(Conf) values, which show how strongly the presence of one drug implies the presence of the other. For example, cocaine appears to have a rather close association with several other drugs in the way that cocaine is often detected with oxycodone, heroin, and benzodiazepines.

 doi:10.20944/preprints202509.1193.v1

Higher confidence and lift values demonstrate a set of drugs that have a stronger relationship to one another; they are more likely to exist together more independently than would be expected to occur by chance. This network-style figure highlights the primary groups of frequently co-used medications, providing meaningful information about patterns of substance use.

In Figure 13, the nodes of the graph represent drugs, and the edges represent rules connecting two drugs with a lift value less than 1, showing negative association rules from the Apriori algorithm. The lift and confidence (Conf) values for each edge identify associations that are weaker or even inverse relationships meaning the presence of one drug could even contribute to reducing the likelihood of the other. For example, the edge connecting cocaine and fentanyl has a relatively low lift value (0.96), indicating a lesser-than-expected co-occurrence. This utilizes a map function to demonstrate alternative, uncommon patterns of use by mapping those negative associations to indicate drug pairs that are less likely to happen together than would be expected by chance.

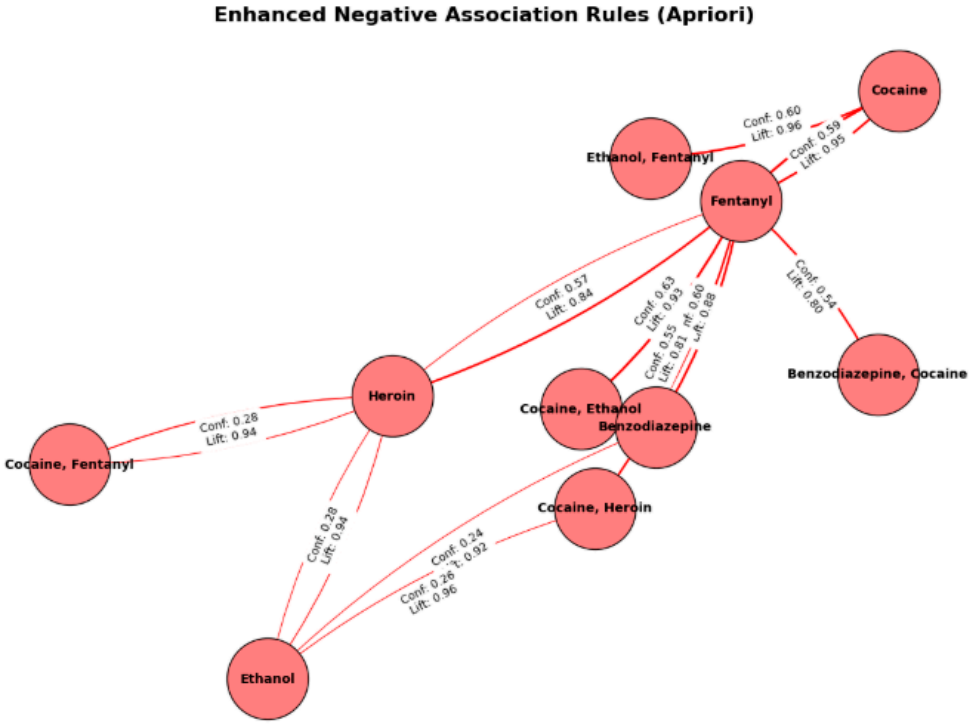

Figure 13. Negative Association Rules (Apriori) visualized.

The network graph shown in Figure 14, represents the positive association rules that were derived from the FP-Growth algorithm. The nodes shown in green are drugs or combinations of drugs in the data set and each link shows the rule connection identified. Values of lift and confidence (Conf) show how closely these drugs co-occur. For instance, cocaine appears to connect with a number of substances (e.g., 3. Oxycodone, Heroin, and benzodiazepenes), suggesting those drugs are used together frequently. The high lift and confidence values between fentanyl and its analog also shows strong correlation. This graph is helpful in showing us the selected important drug co-occurrences by showing the values visually to support public health initiatives and clinical actions intended to reduce risks with co-use.

This network diagram shows in Figure 15, drug pairs with less than one lift value, indicating the negative association rules generated with the FP-Growth algorithm. The edges connecting each red node to each other represent a weaker or inverse relationship. Lift values below 1 demonstrate that the likelihood of the other drug when first given one drug does not increase but may even decrease it. The lift and confidence (Conf) values on each edge show the chance of co-occurrence of two drugs against the chance of co-occurrence at random. For example, a lift of 0. 94 for cocaine and fentanyl, together is less common than would otherwise be expected. Visualizing these adverse associations

 doi:10.20944/preprints202509.1193.v1

like this can assist in documenting drug combinations that are unlikely to occur together, contributing to our knowledge of less frequent use patterns, and this can be directed toward clinical and public health intervention strategies. The results are both the same for both algorithms and that is not the mistake that we made, that is because we provided the same data set and same parameters for both algorithms. We ensured that one algorithm did not have "an advantage" over the other, and that is why we got exactly the same results for both. If we had provided different parameters then we could have also gotten different results as well, and clearly if we had provided different data sets we would have got different results too.

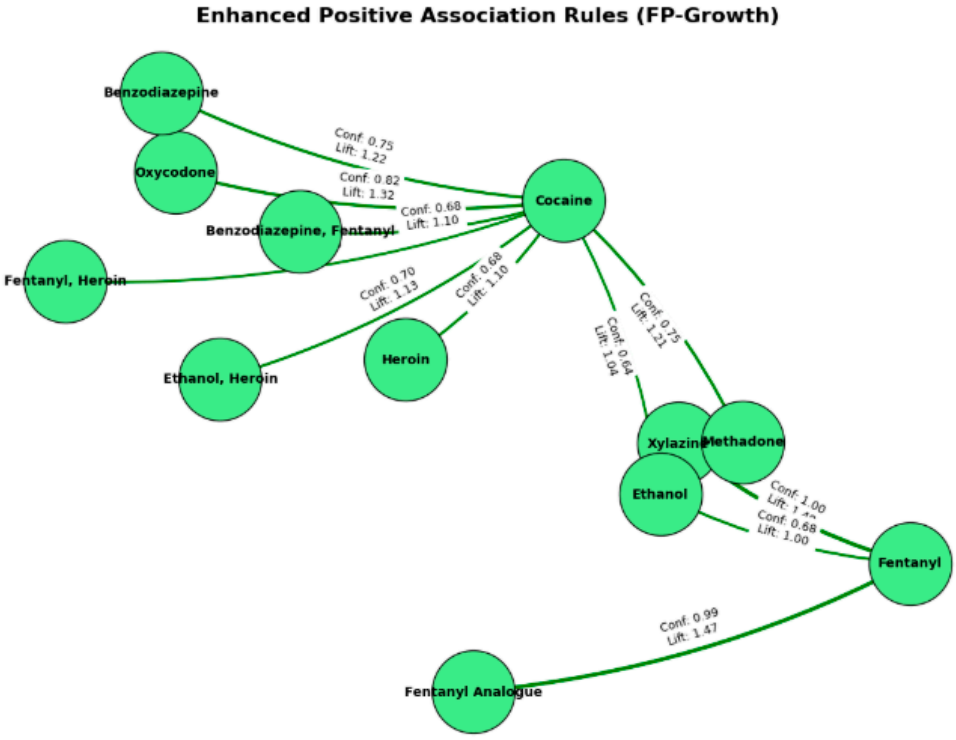

Figure 14. Positive Association Rules (FP-Growth) visualized.

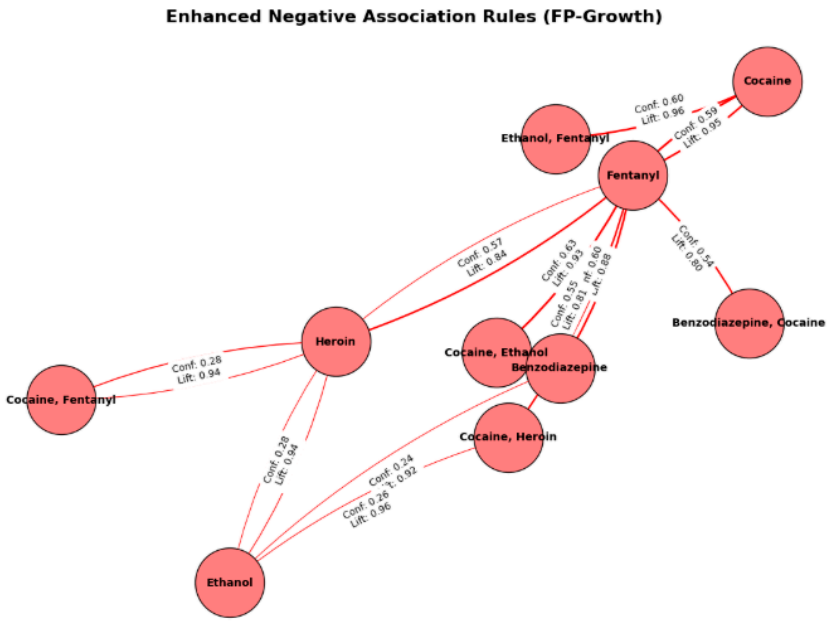

Figure 15. Negative Association Rules (FP-Growth) visualized.

 doi:10.20944/preprints202509.1193.v1

6. The Benefits and Drawbacks of FP-Growth and Apriori Algorithms

Apriori is a simple algorithm that is relatively easy to use on small-to-mid datasets, is well understood and often referenced for generating association rules. Apriori's main drawback when working with large datasets is that it requires multiple scans of the database which can increase computational demands and create inefficiency. Also, Apriori tends to generate many candidate itemsets which can lengthen processing times and lead to unnecessary backs and forths during processing. However, FP-Growth is a scalable and more efficient, designed to work with large datasets. FP-Growth is a faster method than Apriori, as it uses a compressed FP-Tree structure that removes the need for multiple scans of the database, increases processing speed, increases candidate generation. FP-Growth has these advantages, but it is more complicated to construct and uses more memory to build the tree. Additionally, it is important to consider that if most of the dataset. does not fit well into the FP-tree, FP-growth can become disadvantageous in some cases.

The Table below discusses the differences between the two algorithms in terms of features and drawbacks.

Table 4. Compare and Contrast Analysis of Apriori and FP-Growth Algorithms (Patil & Patil, 2022) (Patel & MOHAMMED, 2023) (Garg & Gulia, 2015).

Feature	Apriori Algorithm	FP-Growth Algorithm
Approach	Generates potential itemsets by performing multiple database searches.	Effectively mines patterns and stores data using a tree structure (FP-Tree).
Efficiency	Computationally costly as a result of frequent database searches.	Less frequent database scans making it more efficient.
Memory Usage	Large candidate itemsets and numerous scans result in increased memory requirements	More memory-efficient because an FP-Tree is used to compress the data.
Performance on Large Datasets	Slower and less scalable as a result of the candidate itemsets exponential growth.	Quicker and more scalable because there are fewer candidates being generated.
Ease of Implementation	Simple and easy to understand but can be slow for large datasets	Because of its tree-based structure it is more difficult to implement.
Candidate Generation	Creates candidate itemsets explicitly resulting in redundancy	By keeping common patterns in an FP-Tree candidate generation is avoided.
Use Case	Ideal for datasets that are small to medium in size	Ideal for high-dimensional sizable datasets.
Dependency on Support Threshold	In order to prevent excessive computation, the minimum support must be carefully chosen	Because of tree-based processing it is less sensitive to changes in the support threshold.
Drawback	Slow for large datasets and computationally expensive	Certain data distributions may make complex tree structures challenging to construct.

7. Suggestions and Recommendations based on Findings

Our research reveals important recommendations:

- Policy Implementation: Additional prescription monitoring by focusing on high-risk combinations of drugs (e.g., fentanyl with cocaine, xylazine, ethanol, and/or fentanyl analogues). Increase stigma, enforcement, and barriers to prescribing.

 doi:10.20944/preprints202509.1193.v1

- Public Awareness: Increase public education to inform societal members about polysubstance use and the importance of following medical recommendations.
- Algorithm Use: Use Apriori for introductory analysis but preferably opt for FP-Growth if the dataset is large- FP-Growth is more efficient and scales better.
- We can reduce overdose cases with some combination of prescribing restrictions and public education.

8. Conclusions

Throughout this research, both FP-Growth and Apriori algorithms have been successful for understanding the data and capturing meaningful patterns in opioid and related drug co-occurrence. For instance, FP-Growth and Apriori identified strong positive correlations with several opioids and related drugs, and demonstrated frequent co-occurrence with cocaine, along with heroin, oxycodone, methadone, and benzodiazepines. The negative correlations, conversely, highlighted specific pairs of drugs that may co-occur less than would otherwise be expected. FP-Growth generally provides higher capacity for larger datasets (e.g., G.G. 11981 rows), while Apriori will still remain useful for initial, exploratory research. For high-risk combinations (e.g., fentanyl with ethanol, xylazine, cocaine, and fentanyl analogs), this information contributes to the importance of improved prescription monitoring efforts. It supports targeted public health interventions and educational initiatives addressing polysubstance use. Integrating real-time data analytics and expanded datasets with holistic demographic and contextual variables will allow further refinement of these findings and eventually translate into successful policies and strategies to eliminate substance use-related harm.

References

1. Garg, R., & Gulia, P. (2015). Comparative Study of Frequent Itemset Mining Algorithms Apriori and FP Growth. *International Journal of Computer Applications*, *126*(4), 8–12. https://doi.org/10.5120/ijca2015906030
2. Patil, M., & Patil, T. (2022). Apriori Algorithm against Fp Growth Algorithm: A Comparative Study of Data Mining Algorithms. *SSRN Electronic Journal*, 1–5. https://doi.org/10.2139/ssrn.4113695
3. Usmani, S. A., Kamran, S. K., Muhammad Zeeshan, Islam, N., & Iqra University. (2021). A Comparative Analysis of Apriori and FP-Growth Algorithms for Frequent Pattern Mining Using Apache Spark. *International Hazar Scientific Research Conference*. https://www.researchgate.net/publication/352292388_A_Comparative_Analysis_of_Apriori_and_FP-Growth_Algorithms_for_Frequent_Pattern_Mining_Using_Apache_Spark
4. Patel, B., & MOHAMMED, S. (2023). Comparative analysis of Apriori Algorithm and Frequent Pattern Growth Algorithm in Association Rule Mining. *International Multi-Disciplinary Engineering Conference*, 1–7. https://www.researchgate.net/publication/373776467_Comparative_analysis_of_Apriori_Algorithm_and_Frequent_Pattern_Growth_Algorithm_in_Association_Rule_Mining
5. P.Naga , K. (2022). Comparative Analysis of Apriori and FP-Growth Algorithms For Frequent Item Sets. *International Journal of Advanced in Management, Technology and Engineering Sciences*, *XII*(IV), 1–18. https://stannscollege.in/ssr/wp-content/uploads/2023/09/Kavitha.pdf?utm_source=chatgpt.com ISSN NO: 2249-7455.
6. Gill, S. H., Razzaq, M. A., Ahmad, M., Almansour, F. M., Haq, I. U., Jhanjhi, N. Z., ... & Masud, M. (2022). Security and privacy aspects of cloud computing: a smart campus case study. Intelligent Automation & Soft Computing, 31(1), 117-128.
7. Attaullah, M., Ali, M., Almufareh, M. F., Ahmad, M., Hussain, L., Jhanjhi, N., & Humayun, M. (2022). Initial stage COVID-19 detection system based on patients' symptoms and chest X-ray images. Applied Artificial Intelligence, 36(1), 2055398.
8. Sarfraz Nawaz Brohi , NZ Jhanjhi , Nida Nawaz Brohi , et al. Key Applications of State-of-the-Art Technologies to Mitigate and Eliminate COVID-19. TechRxiv. April 15, 2020. DOI: 10.36227/techrxiv.12115596.v2

 doi:10.20944/preprints202509.1193.v1

9. Sangkaran, T., Abdullah, A., JhanJhi, N. Z., & Supramaniam, M. (2019). Survey on isomorphic graph algorithms for graph analytics. International Journal of Computer Science and Network Security, 19(1), 85-92.
10. Babbar, H., Rani, S., Masud, M., Verma, S., Anand, D., & Jhanjhi, N. (2021). Load balancing algorithm for migrating switches in software-defined vehicular networks. *Computational Materials and Continua*, *67*(1), 1301-1316.
11. Hall, O. E., Hall, O. T., Eadie, J. L., Teater, J., Gay, J., Kim, M., ... & Noonan, R. K. (2021). Street-drug lethality index: a novel methodology for predicting unintentional drug overdose fatalities in population research. *Drug and alcohol dependence*, *221*, 108637.
12. Gopi, R., Sathiyamoorthi, V., Selvakumar, S., et al. (2022). Enhanced method of ANN based model for detection of DDoS attacks on multimedia Internet of Things. *Multimedia Tools and Applications*, *81*(36), 26739-26757. https://doi.org/10.1007/s11042-021-10640-6
13. Lee, S., Abdullah, A., & Jhanjhi, N. Z. (2020). A review on honeypot-based botnet detection models for smart factory. *International Journal of Advanced Computer Science and Applications*, *11*(6).
14. Zaman, N., Abdullah, A. B., & Jung, L. T. (2011, March). Optimization of energy usage in wireless sensor network using Position Responsive Routing Protocol (PRRP). In *2011 IEEE Symposium on Computers & Informatics* (pp. 51-55). IEEE.
15. M. R. Azeem, S. M. Muzammal, N. Zaman and M. A. Khan, "Edge Caching for Mobile Devices," *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, Karachi, Pakistan, 2022, pp. 1-6, doi: 10.1109/MACS56771.2022.10022729.
16. Humayun, M., Khalil, M. I., Almuayqil, S. N., & Jhanjhi, N. Z. (2023). Framework for detecting breast cancer risk presence using deep learning. *Electronics*, *12*(2), 403.
17. Eldhai, A. M., Hamdan, M., Abdelaziz, A., Hashem, I. A. T., Babiker, S. F., Marsono, M. N., ... & Jhanjhi, N. Z. (2024). Improved feature selection and stream traffic classification based on machine learning in software-defined networks. *IEEE access*, *12*, 34141-34159.