

Discovery of NWC Patterns

Low-level Design



UNIVERSITY OF MINNESOTA
Driven to DiscoverSM

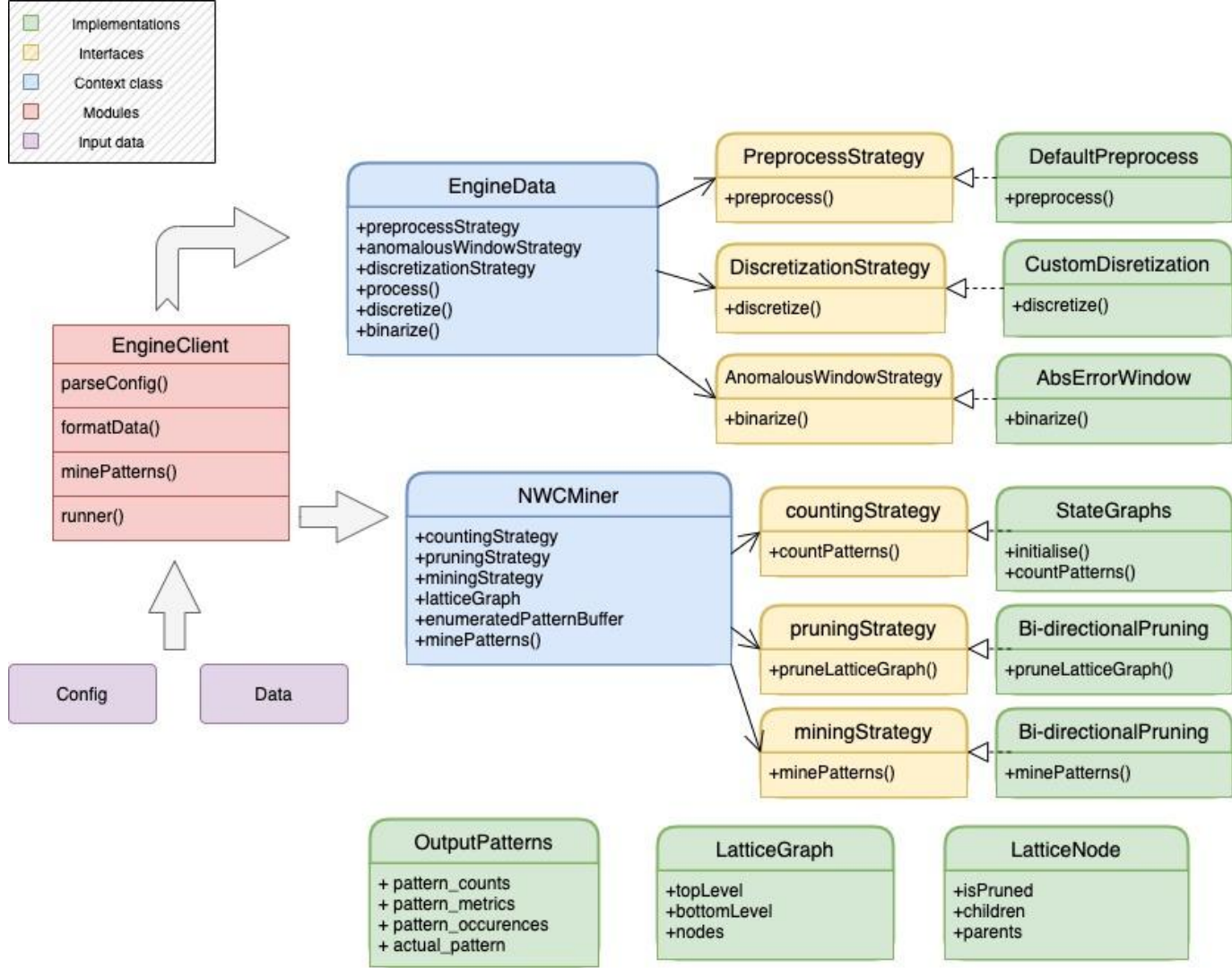
Possible Configurations for Engine NWC detection

- Input Data file path
- Support Threshold
- Cross-k Threshold
- Anomalous Window Length
- Number of bins - Discretization
- Attribute names to analyse for patterns
- Anomalous Target Variable - Anomalous Window Strategy
 - Threshold based - specify attribute name
 - Evaluation based - derive new column from existing columns
- Pattern Length range
- Lag
- Extra parameters - to switch between type of pruning
 - Tight lattice bounds, lattice bounds, support-based

Expected Output

- The Output pattern
- The locations where it exists,
 - or coincides with anomalous windows (within the lag)
- Various counts for pattern
- Support for pattern
- Ripley's Cross-k for pattern
- Confidence measure for pattern

Proposed LLD



- Composition over Inheritance
- Injecting **Strategies** disassociates clients from Family of algorithms
- Makes package more extensible and flexible with inclusion Of new strategies.

Abstractions

- The concept of pass/trip/MET is abstracted within EngineData preprocessing,
 - Along with definition of anomalous Window
 - One way of abstracting multiple passes / MET is by dividing into multiple data frames
 - It might have complications in mapping back to pattern occurrences
- Counting missing iterations would have to be included in StateGraph initialization
 - Thus, it would have to be provided as an option in the generic package

References

- <https://refactoring.guru/refactoring>
- https://www.youtube.com/watch?v=v9ejT8FO-7I&ab_channel=ChristopherOkhravi