CEWP Data Source Audit (Updated)

Spatio-Temporal Resolutions & Imputation Methods

Pipeline Version: January 2026
Target Region: Central African Republic (CAR)
Analysis Grid: H3 Resolution 5 (~10km hexagonal cells)
Grid Coverage: ~3,000 cells (updated from spatial bounds)
Temporal Spine: 14-day intervals aligned to 2000-01-01
Generated: 2026-01-06

Executive Summary

This audit documents all data sources integrated into the Conflict Early Warning Pipeline (CEWP), detailing their native resolutions, pipeline transformations, and imputation strategies for missing values. The pipeline harmonizes **21+ distinct data sources** across **six thematic categories** into a unified analytical framework suitable for conflict prediction modeling.

Category	Sources	Native Resolution	Target Resolution
Environmental	6	30m – 11km, hourly/daily	H3-5, 14-day
Conflict & Events	3	Point, daily	H3-5, 14-day
Socio-Political	4	Admin-1/2/3, annual/quarterly	H3-5, 14-day
Economic	5	National/Market, daily/monthly	National/H3-5, 14-day
Infrastructure	6	Point/polygon, static	H3-5, static
Demographics	1	100m, annual	H3-5, annual

Table 1: Data source categories with resolution transformations

---

1. Environmental Data

Environmental variables are aggregated from Google Earth Engine collections to H3 resolution 5 with 14-day temporal windows using **server-side Map-Reduce architecture**.

1.1 CHIRPS Precipitation

Parameter	Value
Source	UCSB-CHG/CHIRPS/DAILY (via GEE)
Native Spatial	0.05° (~5.5km)
Native Temporal	Daily
Aggregation	H3-5 zonal mean, 14-day sum
Output Column	precip_mean_depth_mm → chirps_precip_anomaly
Imputation	Forward-fill (limit=4 steps / 56 days)
Transformation	Anomaly vs 6-period rolling baseline
Coverage	1981–present

1.2 ERA5-Land Climate

Parameter	Value
Source	ECMWF/ERA5_LAND/HOURLY (via GEE)
Native Spatial	0.1° (~11km)
Native Temporal	Hourly
Variables	Temperature (2m), Soil Moisture (Layer 1)
Aggregation	H3-5 zonal mean, 14-day mean
Output Columns	temp_mean → era5_temp_anomaly soil_moisture_mean → era5_soil_moisture_anomaly
Imputation	Forward-fill (limit=4 steps)
Transformation	Anomaly vs climatological baseline
Coverage	1950–present (~5-day lag)

1.3 MODIS Vegetation (NDVI)

Parameter	Value
Source	MODIS/061/MCD43A4 (via GEE)
Native Spatial	500m
Native Temporal	Daily (16-day composite)
Aggregation	H3-5 zonal max, 14-day max
Output Column	`ndvi_max` → `ndvi_anomaly`
Imputation	Forward-fill (limit=4 steps)
Transformation	Anomaly vs 6-period rolling mean
Notes	NDVI = (NIR - Red) / (NIR + Red); range [-1, 1]

1.4 VIIRS Nighttime Lights

Parameter	Value
Source	NASA/VIIRS/002/VNP46A2 (via GEE)
Native Spatial	500m
Native Temporal	Daily
Aggregation	H3-5 zonal mean, 14-day mean
Output Column	`ntl_mean` → `nightlights_intensity`
Imputation	Forward-fill (limit=4 steps)
Coverage	2012–present
Notes	DNB_BRDF_Corrected_NTL band; pre-2012 years have NULL
Structural Break	NULL before 2012 (sensor launch date)

1.5 JRC Global Surface Water

Parameter	Value
Source	JRC/GSW1_4/MonthlyHistory (pre-2022) Landsat 8/9 (2022+)
Native Spatial	30m
Native Temporal	Monthly (JRC) / Daily composite (Landsat)
Aggregation	H3-5 zonal mean + max, 14-day
Output Columns	water_local_mean → water_coverage_lag1 water_local_max → water_presence_lag1
Imputation	Forward-fill (limit=4 steps)
Transformation	Lag 1 step (14 days)
Notes	Binary water detection (MNDWI > 0.1 for Landsat)

---

2. Conflict & Event Data

Event data is aggregated from point locations to H3 cells with 14-day temporal windows. **Zero-fill imputation** is used under the assumption that no recorded event equals no event occurrence.

2.1 ACLED (Armed Conflict Location & Event Data)

Parameter	Value
Source	Local CSV + API (data/raw/ACLED.csv)
Native Spatial	Point (lat/lon, geo_precision 1-3)
Native Temporal	Daily (event_date)
Aggregation	H3-5 point-in-polygon, 14-day sum
Output Columns	fatalities → fatalities_14d_sum, fatalities_1m_lag fatalities → conflict_density_10km (decay_30d)

Parameter	Value
	protest_count, riot_count (by event_type) regional_risk_score_lag1 (admin-level aggregation)
Imputation	Zero-fill for missing (no events = 0)
Transformations	Lag (1 step), Decay (30d half-life), Regional aggregation
Coverage	1997–present (CAR: 2000+)
Volume	~7,500+ events in CAR
Geo Precision Split	Precision 1-2: Local features Precision 3: Regional features only

2.2 GDELT (Global Database of Events, Language, and Tone)

Parameter	Value
Source	BigQuery gdelt-bq.gdeltv2.events
Native Spatial	Point (Actor1Geo_Lat/Lon)
Native Temporal	Daily (SQLDATE)
Aggregation	H3-5 point-in-polygon, 14-day sum/mean
Output Columns	gdelt_event_count → events_3m_lag (decay_90d) gdelt_event_count → gdelt_decay_30d Optional: gdelt_goldstein_mean, gdelt_avg_tone
Imputation	Zero-fill for missing
Transformations	Decay (30d and 90d half-lives)
Coverage	2015–present (v2)
Notes	Uses country code "CF" for CAR; cached to reduce costs

2.3 CrisisWatch NLP Analysis (NEW)

Parameter	Value
Source	International Crisis Group CrisisWatch monthly reports
Native Spatial	Country/Admin-1 mentions with spatial confidence scores
Native Temporal	Monthly reports
Aggregation	Topic-based spatial disaggregation to H3-5 Weighted by spatial confidence score
Output Columns	`crisiswatch_topic_{0-9}` (10 topics via NLP)
Imputation	Zero-fill for missing topics
Transformation	Pivot table with spatial_confidence as aggregation weight
Coverage	2018–present
Notes	Text-based risk indicators; complements event databases

3. Socio-Political Data

Socio-political data requires **spatial disaggregation** from administrative boundaries to H3 cells. These sources capture ethnic power dynamics, displacement patterns, and food security conditions.

3.1 EPR (Ethnic Power Relations)

Parameter	Value
Source	ETH Zurich EPR-2021 (Core CSV + GeoEPR polygons)
Native Spatial	Ethnic group polygon (GeoEPR)
Native Temporal	Annual (from-to year ranges)
Aggregation	Polygon → H3-5 via `polygon_to_cells()`, annual

Output Columns	`ethnic_group_count`
	`epr_excluded_groups_count`
	`epr_discriminated_groups_count`

Parameter	Value
	epr_status_mean ⚠ **MISSING:** epr_horizontal_inequality (Gini)
Imputation	Zero-fill for cells outside any ethnic polygon
Coverage	1946–2021 (expanded year-by-year)
Post-2021	Uses 2021 data (acceptable for slow-changing features)
Notes	27 polygons in CAR; requires calculation of Gini coefficient

3.2 IOM DTM (Displacement Tracking Matrix)

Parameter	Value
Source	IOM DTM API v3
Native Spatial	Admin-2 (sub-prefecture) or Admin-1
Native Temporal	Survey rounds (irregular, ~quarterly)
Aggregation	Admin-2 → H3-5 via area-weighted density, 14-day
Output Column	iom_displacement_sum → iom_displacement_count_lag1 iom_data_available (structural break flag)
Imputation	Zero-fill between survey rounds
Transformation	Lag 1 step
Coverage	2014–present
Notes	Fallback to Admin-1 if Admin-2 unavailable Irregular temporal coverage requires zero-fill strategy

3.3 FEWS NET IPC (Food Security Phases)

Parameter	Value
Source	FEWS NET Data Warehouse API
Native Spatial	Admin-1 (prefecture)

Parameter	Value
Native Temporal	Quarterly projections
Aggregation	Admin-1 → H3-5 via polygon overlay, 14-day max
Output Column	ipc_phase_class (1-5 scale) → ipc_phase_class_lag1
Imputation	Constant=0 before 2009-01-01 (pre-IPC era) Forward-fill after (limit=4 steps)
Transformation	Lag 1 step
Coverage	2009–present
Notes	Requires FEWS_NET_TOKEN in .env

3.4 WFP Food Prices (UPDATED - Local Market Level)

Parameter	Value
Source	WFP VAM API + FEWS NET Markets
Native Spatial	Market point locations (10 markets in CAR)
Native Temporal	Monthly (irregular reporting)
Aggregation	Market → Nearest H3-5 via Voronoi tessellation
Output Columns	price_maize, price_rice, price_oil, price_sorghum price_maize_shock, price_rice_shock, price_oil_shock, price_sorghum_shock
Imputation	Forward-fill (limit=4 steps), then zero-fill
Transformation	Shock = current / rolling_mean(12 months)
Coverage	2015–present
Commodities	Maize (Corn), Rice (Milled), Palm Oil, Sorghum
Notes	Spatial mapping via cKDTree nearest neighbor

4. Economic Data (EXPANDED COVERAGE)

Economic indicators include both **global macro indicators** (non-spatial) and **local market prices** (spatially distributed).

4.1 Macro-Economic Indicators (Yahoo Finance)

Parameter	Value
Source	yfinance Python package
Tickers	GC=F (Gold Futures) CL=F (Oil Futures) ^GSPC (S&P 500) EURUSD=X (EUR/USD FX)
Native Spatial	Global (non-spatial)
Native Temporal	Daily (trading days)
Aggregation	National-level broadcast to all H3 cells, 14-day mean

Output Columns	`gold_price_usd` → `gold_price_usd_lag1`
	`oil_price_usd` → `oil_price_usd_lag1`
	`sp500_index` → `sp500_index_lag1`
	`eur_usd_rate` → `eur_usd_rate_lag1`

Imputation	Forward-fill for non-trading days (weekends, holidays)
Transformation	Lag 1 step (14 days)
Coverage	2000-08-30 (Gold futures) to present
Storage	Dedicated `economic_drivers` table

4.2 Local Market Prices

See Section 3.4 (WFP Food Prices) - now categorized under Socio-Political but includes economic shock features.

---

5. Infrastructure & Geography (Static Features)

Static infrastructure features are computed **once** and joined to all temporal observations. Distance calculations use **cKDTree** for efficient nearest-neighbor queries in UTM projection (EPSG:32634).

Data Source	Native Resolution	Output Features	Notes
Copernicus DEM 90m	90m raster	`elevation_mean`, `slope_mean`, `terrain_ruggedness_index`	Via GEE server-side
GRIP4 Roads	Polyline network	`dist_to_road` (km)	PBL Region 3 (Africa)
HydroRIVERS	Polyline network	`dist_to_river` (km)	Stream order ≥ 3

IPIS Mining Sites	Point locations	`dist_to_diamond_mine`
		`dist_to_gold_mine`
		`dist_to_large_mine`
		`dist_to_controlled_mine`
		`dist_to_large_gold_mine`
			914 sites in CAR

OSM Settlements	Point locations	`dist_to_city` `dist_to_capital`	Via HDX CAR dataset
GADM Boundaries	Polygons	Admin-level identifiers for regional aggregation	Used for regional_risk_score

Table 2: Static infrastructure data sources and derived features

Distance Calculation Method:

```python
# UTM Zone 34N projection for metric distances
grid_points = np.fliplr(get_h3_centroids(unique_h3))  # lon, lat
feature_points = gdf[['longitude', 'latitude']].values
tree = cKDTree(feature_points)
distances, _ = tree.query(grid_points, k=1)
distances_km = distances / 1000.0  # Convert meters to km
```

Sentinel Values: If a feature (e.g., `dist_to_controlled_mine`) is 100% NULL due to no features of that type existing, the pipeline fills with a large sentinel value (500 km) and adds a missingness flag (`dist_to_controlled_mine_is_missing`).

6. Demographics

6.1 WorldPop Population

Parameter	Value
Source	WorldPop 100m constrained (R2025A)
Native Spatial	100m raster
Native Temporal	Annual
Aggregation	H3-5 zonal sum, annual
Output Columns	`pop_count` → `pop_log` (log1p transformation) `is_worldpop_v1` (structural break indicator)
Imputation	Forward-fill within year Backward extrapolation (2.5% annual growth) pre-2015
Coverage	2000–2030 (projections 2026-2030)
File Variants	`caf_ppp_{year}_UNadj.tif` (2000-2014) `caf_pop_{year}_CN_100m_R2025A_v1.tif` (2015+)
Structural Break	2015: V1 (census-adjusted) → V2 (constrained model)

⚠ **Important:** The structural break at 2015 represents a methodology change:

- **Pre-2015 (V1):** Census-based adjustment
- **Post-2015 (V2):** Constrained model with built environment data

The `is_worldpop_v1` flag enables the model to learn different relationships for each period.

---

7. Temporal Context Features (NEW - Added in Phase 5)

7.1 Seasonal Features

Parameter	Value
Source	Derived from temporal spine `date` column
Output Columns	`month_sin` = sin(2π × month / 12) `month_cos` = cos(2π × month / 12) `is_dry_season` = {1 if month ∈ {11,12,1,2,3,4}, else 0}
Rationale	Cyclical encoding captures seasonality Dry season flag captures operational constraints (road access)
Coverage	All time periods
Notes	Added during spine creation; NOT in features.yaml registry **Action Required:** Add to registry for documentation

---

8. Imputation Strategy (Updated)

Missing values are handled through **feature-specific strategies** that preserve temporal and spatial coherence while avoiding information leakage from future observations.

8.1 Default Strategy

```yaml
imputation:
  defaults:
    method: "forward_fill"
    limit: 4  # 4 steps = 56 days maximum gap
```

The default imputation method is **forward-fill with a 4-step (56-day) limit**. This preserves the last known value while preventing stale data from persisting indefinitely.

8.2 Feature-Specific Overrides

Feature Category	Method	Details
Population	Backward extrapolation	2.5% annual growth rate for years < 2015
IPC Phase	Constant then forward-fill	Value=0 before 2009-01-01 (pre-IPC era); forward-fill after
Conflict events	Zero-fill	No event recorded = 0 fatalities/protests/riots
GDELT events	Zero-fill	No event = 0 count
IODA outages	Zero-fill	No outage = 0
IOM displacement	Zero-fill	Between survey rounds = 0 new displacement
EPR features	Zero-fill	Cells outside ethnic polygons = 0
Economic prices	Forward-fill	Tolerant: limit=8 steps for macro indicators Strict: limit=4 steps for local prices
Environmental	Forward-fill	Strict: limit=4 steps; never extrapolate beyond 56 days
Distance features	Sentinel value	If 100% NULL, fill with feature-specific sentinel (50-500 km) + missingness flag

Table 3: Feature-specific imputation overrides

8.3 Structural Break Handling

Several features have **known structural breaks** that require special handling:

Feature	Break Date	Handling
VIIRS Nightlights	2012-01-01	NULL before (sensor launch); zero-fill
WorldPop	2015-01-01	V1→V2 methodology; add `is_worldpop_v1` flag
IOM DTM	2014-01-01	No data before; zero-fill; add `iom_data_available` flag
IPC	2009-01-01	No classification before; constant=0
EPR	2021-12-31	No updates after; use 2021 data (acceptable for slow-changing)

9. Resolution Transformations (Updated)

All data sources are harmonized to a **common spatio-temporal framework** through aggregation or disaggregation.

9.1 Spatial Resolution Mapping

Native Resolution	Sources	Transformation	Method
30m	Landsat water	Zonal statistics → H3-5	GEE server-side
90m	Copernicus DEM	Zonal statistics → H3-5	Rasterio + H3
100m	WorldPop	Zonal sum → H3-5	Rasterio + H3
500m	MODIS, VIIRS	Zonal statistics → H3-5	GEE server-side
5km	CHIRPS	Zonal mean → H3-5	GEE server-side
11km	ERA5	Zonal mean → H3-5	GEE server-side
Point	ACLED, GDELT, mines, settlements	Point-in-polygon → H3-5	GeoPandas sjoin
Admin-1	IPC, IODA	Polygon overlay → H3-5	Area-weighted or max
Admin-2	IOM	Area-weighted disaggregation → H3-5	Population density weights
Admin-3	ACLED regional	Polygon overlay → H3-5	For regional_risk_score

Table 4: Spatial resolution transformations to H3 Resolution 5 (~253 km² cells)

9.2 Temporal Resolution Mapping

Native Frequency	Sources	Transformation	Method
Hourly	ERA5	14-day mean	GEE server-side
Daily	CHIRPS, ACLED, GDELT, Landsat	14-day sum/mean/max	Binning by spine_date

Native Frequency	Sources	Transformation	Method
16-day composite	MODIS	14-day max	Overlapping windows
Monthly	JRC water, WFP prices	14-day interpolation	Forward-fill within month
Quarterly	IOM, IPC	14-day forward-fill	Limit=4 steps
Annual	WorldPop, EPR	Joined to all steps in year	Broadcast annual → 14-day
Static	DEM, roads, rivers, mines	Joined to all time steps	Cross join

Table 5: Temporal resolution transformations to 14-day intervals

---

10. Known Data Gaps & Issues (Updated)

Issue	Impact	Status	Priority
EPR horizontal_inequality missing	Feature in models.yaml but no generation code	❌ Critical	**HIGH**
Seasonal features not in registry	month_sin, month_cos, is_dry_season undocumented	⚠️ Documentation	**MEDIUM**
Distance features not in registry	All dist_* features missing from features.yaml	⚠️ Documentation	**MEDIUM**
FEWS_NET_TOKEN missing	IPC + market prices = NULL for some users	⚠️ User setup	**MEDIUM**
Market locations CSV BOM	Encoding issue caused 0 markets loaded	✅ Fixed	**CLOSED**
VIIRS pre-2012	NTL = NULL	✅ Expected	**CLOSED**
EPR post-2021	Uses 2021 data for 2022-2025	✅ Acceptable	**CLOSED**
IOM survey gaps	Irregular temporal coverage	✅ Zero-filled	**CLOSED**
h3 API deprecation	h3.k_ring → h3.grid_disk	✅ Migrated	**CLOSED**

Table 6: Known data gaps and their resolution status

---

11. Temporal Transformations (Detailed)

Raw features undergo temporal transformations to capture **lagged effects**, **cumulative impacts**, and **anomalies** relative to historical baselines.

Transformation	Formula	Purpose	Example Output
lag_1_step	x(t-1)	Prevent leakage; capture delayed effects	temp_mean_lag1
sum_1_step	Σ x(t) over window	Accumulate events within period	fatalities_14d_sum
decay_30d	EWM(x, halflife=2.14 steps)	Weight recent events more heavily	fatalities_decay_30d
decay_90d	EWM(x, halflife=6.43 steps)	Long-term conflict persistence	events_3m_lag
anomaly_6_step	x(t) - mean(t-6:t)	Deviation from rolling baseline	chirps_precip_anomaly
shock_12m	x(t) / mean(t-12m:t)	Price spike detection	price_maize_shock
spatial_mean_k1	mean(h3.grid_disk(k=1))	Neighborhood context	temp_spatial_mean_k1

Table 7: Temporal transformation functions applied during feature engineering

Transformation Implementation Details

Decay Functions:

```python
# 30-day half-life (2.14 steps at 14-day resolution)
spine['fatalities_decay_30d'] = spine.groupby('h3_index')['fatalities'].ewm(halflife=2.14).mean()

# 90-day half-life (6.43 steps)
spine['events_3m_lag'] = spine.groupby('h3_index')['gdelt_event_count'].ewm(halflife=6.43).mean()
```

Anomaly Functions:

```python
# 6-period (84-day) rolling baseline
rolling_mean = spine.groupby('h3_index')['precip_mean_depth_mm'].rolling(window=6, min_periods=1).mean()
spine['chirps_precip_anomaly'] = spine['precip_mean_depth_mm'] - rolling_mean
```

Shock Functions:

```python
# 12-month (26-step at 14-day resolution) rolling baseline
window = int(12 * (30.0 / 14.0))  # ~26 steps
rolling_mean = spine.groupby('h3_index')['price_maize'].rolling(window=window, min_periods=1).mean()
spine['price_maize_shock'] = spine['price_maize'] / (rolling_mean + 1e-6)
```

---

12. Target Variable Construction

The pipeline creates **multiple target variables** for different prediction tasks and horizons.

12.1 Target Definitions

Target	Formula	Purpose	Horizon
target_1_step	LEAD(fatalities_14d_sum, 1)	14-day forecast	1 step ahead
target_2_step	LEAD(fatalities_14d_sum, 2)	1-month forecast	2 steps ahead
target_6_step	LEAD(fatalities_14d_sum, 6)	3-month forecast	6 steps ahead

SQL Implementation:

```sql
-- In build_feature_matrix.py::build_dynamic_query()
LEAD(t.fatalities_14d_sum, {steps})
  OVER (PARTITION BY t.h3_index ORDER BY t.date)
  as target_{steps}_step
```

12.2 Target Quality Metrics

Expected target characteristics:

- **Null Rate:** <20% (only at end of time series where LEAD fails)
- **Positive Rate:** 1-5% (conflict is rare)
- **Non-negative:** All values ≥ 0 (fatalities cannot be negative)

Validation Query:

```sql
SELECT
    COUNT(*) FILTER (WHERE target_1_step IS NULL) * 100.0 / COUNT(*) as null_pct,
    COUNT(*) FILTER (WHERE target_1_step > 0) * 100.0 / COUNT(*) as positive_pct,
    COUNT(*) FILTER (WHERE target_1_step < 0) as negative_count
FROM car_cewp.feature_matrix;
-- Expected: null_pct ~5-10%, positive_pct ~2-4%, negative_count = 0
```

---

13. Feature Count Summary

By Source Category

Category	Raw Columns	Transformed Features	Total
Environmental	8	16 (with anomalies, lags)	24
Conflict	5	15 (with decays, lags, regional)	20
Economic	8	12 (with lags, shocks)	20
Socio-Political	8	4 (with lags)	12
Infrastructure	12	0 (static)	12
Demographics	2	3 (with log, structural break)	5
Temporal Context	0	3 (seasonal)	3
Total	**43**	**53**	**96**

Feature Registry Coverage

Status	Count	Percentage
✅ Fully documented in features.yaml	65	68%
⚠️ Generated but not in registry	28	29%
❌ Declared but not implemented	3	3%

Action Items:

1. Add seasonal features to features.yaml registry
2. Add all distance features to features.yaml registry
3. Implement EPR horizontal_inequality calculation
4. Add NLP topic features to registry

---

14. Data Validation Checkpoints

The pipeline should enforce these assertions at key stages:

Stage 1: Post-Ingestion

```sql
-- Check H3 type consistency
SELECT table_name, data_type
FROM information_schema.columns
WHERE table_schema = 'car_cewp' AND column_name = 'h3_index'
  AND data_type NOT IN ('bigint', 'int8');
-- Must return 0 rows

-- Check temporal coverage
SELECT
    MIN(date) as earliest,
    MAX(date) as latest,
    COUNT(DISTINCT date) as unique_dates
FROM car_cewp.temporal_features;
-- Expected: ~180 dates per year (26 steps × 7 years = 182)
```

Stage 2: Post-Feature Engineering

```sql
```

```
-- Check for missing critical features
SELECT
    COUNT(*) FILTER (WHERE chirps_precip_anomaly IS NULL) as precip_nulls,
    COUNT(*) FILTER (WHERE fatalities_14d_sum IS NULL) as fatality_nulls,
    COUNT(*) FILTER (WHERE pop_log IS NULL) as pop_nulls
FROM car_cewp.temporal_features;
-- All should be < 5% of total rows

-- Check feature value ranges
SELECT
    MIN(elevation_mean) as min_elev,
    MAX(elevation_mean) as max_elev,
    MIN(dist_to_road) as min_road_dist,
    MAX(dist_to_road) as max_road_dist
FROM car_cewp.features_static;
-- min_elev should be ~300m, max_elev ~1400m (CAR topography)
-- max_road_dist should be < 100km
```

Stage 3: Pre-Modeling

```
python

# Check target distribution
df = pd.read_parquet('data/processed/feature_matrix.parquet')

for col in ['target_1_step', 'target_2_step', 'target_6_step']:
    null_rate = df[col].isna().mean()
    positive_rate = (df[col] > 0).mean()

    assert null_rate < 0.20, f"{col} null rate too high: {null_rate}"
    assert 0.001 < positive_rate < 0.10, f"{col} prevalence out of range: {positive_rate}"
```

---

15. Pipeline Execution Summary

Typical Execution Times (Full Historical Run 2000-2025)

Phase	Duration	Bottleneck
Static Ingestion	30-45 min	DEM download + terrain calculation

Phase	Duration	Bottleneck
Dynamic Ingestion	2-4 hours	GEE environmental data (25 years)
Feature Engineering	45-90 min	Spatial joins + temporal transforms
Feature Matrix Build	15-30 min	SQL query + type enforcement
Model Training	30-60 min	XGBoost hyperparameter tuning
Total	**4-7 hours**	GEE API rate limits

Data Volume

Artifact	Typical Size	Rows
`features_static`	~500 KB	~3,000
`temporal_features`	~200-500 MB	~450,000 (3,000 cells × 150 dates)
`feature_matrix.parquet`	~100-300 MB	~450,000
`acled_events`	~2 MB	~7,500 events
`environmental_features`	~1-2 GB	~450,000

Appendix A: Feature Registry (Complete List)

See separate document: `Feature_Registry_Complete.xlsx`

Or query from code:

```python
import yaml
with open('configs/features.yaml') as f:
    features = yaml.safe_load(f)

for item in features['registry']:
    print(f"{item['output_col']}: {item['source']} → {item['transformation']}")
```

Appendix B: Database Schema Diagrams

```
car_cewp schema:
│
├── features_static (spatial foundation)
│   └── Columns: h3_index (PK), geometry, elevation_mean, slope_mean,
│       terrain_ruggedness_index, dist_to_*, admin1, admin2, admin3
│
├── temporal_features (time series)
│   └── Columns: h3_index, date (composite PK),
│       [environmental], [conflict], [economic], [socio-political]
│
├── Raw ingestion tables:
│   ├── acled_events (h3_index, event_date, fatalities, event_type, ...)
│   ├── environmental_features (h3_index, date, precip_mean_depth_mm, ...)
│   ├── economic_drivers (date, gold_price_usd, oil_price_usd, ...)
│   ├── food_security (date, market, commodity, value)
│   ├── iom_displacement_h3 (h3_index, date, iom_displacement_sum)
│   └── ipc_h3 (h3_index, date, ipc_phase_class)
│
└── Spatial reference tables:
    ├── population_h3 (h3_index, year, pop_count)
    ├── grip4_roads_h3 (h3_index, road_density)
    ├── geoepr_polygons (group_id, status, geometry)
    └── market_locations (market_id, latitude, longitude)
```

Document Change Log

Version	Date	Changes
1.0	2025-12-11	Initial release

Version	Date	Changes
2.0	2026-01-06	**Major update:** • Added macro-economic indicators (Section 4.1) • Added CrisisWatch NLP (Section 2.3) • Added seasonal features (Section 7) • Updated imputation strategies • Added EPR missing feature documentation • Expanded validation checkpoints • Added execution time benchmarks

Generated: 2026-01-06
Auditor: CEWP Development Team
Pipeline Version: January 2026 (Phase 5 Complete)
Next Review: After next major feature addition or data source integration