ORNL/TM-2023/3047

Oak Ridge National Laboratory USA Structures Phase 2 Technical Report Version 1

Taylor Hauser
Jessica Moehl
Erik Schmidt
Daniel Adams
Matthew Whitehead
Bennett Morris
H. Lexie Yang

August 2023

OAK RIDGE National Laboratory

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2023/3047

Geospatial Science and Human Security Division

USA STRUCTURES PHASE 2

Taylor Hauser
Jessica Moehl
Erik Schmidt
Daniel Adams
Matthew Whitehead
Bennett Morris
H. Lexie Yang

August 2023

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-Battelle LLC
for the

US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API	Application Programming Interface
EMD	Expected Mean Distance
ETL	Extract Transform and Load
FAA	Federal Aviation Administration
FEMA	Federal Emergency Management Agency
GAUNTLET	Geographic Augmentation of Extracted Building Features Tool
HIFLD	Homeland Infrastructure Foundation Level Data
HUD	Department of Housing and Urban Development
ISO	Isolation Forest
LOF	Local Outlier Factor
NAD	National Address Database
NGA	National Geospatial Intelligence Agency
NNI	Nearest Neighbor Index
OMD	Observed Mean Distance
OSM	OpenStreetMap
ORNL	Oak Ridge National Laboratory
REST	Representational State Transfer
SA	Scale Analysis
SGDOSVM	Stochastic Gradient Descent One-Class Support Vector Machine
UUID	Universally Unique IDentifier

1. INTRODUCTION

Spatially accurate data of critical infrastructures are essential to effective disaster preparedness, response, and recovery. Precise location and building outlines provide the most accurate data for characterizing impacts of hazards and effectively serve response, recovery, and mitigation efforts, as well as the people affected by the disaster. Since 2017, Oak Ridge National Laboratory (ORNL) has partnered with the Federal Emergency Management Agency (FEMA) to establish a comprehensive and open source national database of building footprints called USA Structures. Several key attributions have been added to the dataset to support rapid disaster response. In the most recent update to the dataset, ORNL developed two additional attributions to the structures, leveraging several authoritative data sources.

2. STRUCTURE OCCUPANCY CLASSIFICATION

The use of a structure is a critical attribute for a wide variety of analyses. For example, emergency response, population modeling, and risk assessments all benefit from knowing the general use of a structure. ORNL aims to meet the needs of the emergency response, national security, and scientific communities by filling two attributes in the USA Structures schema: OCC_CLS and PRIM_OCC.

The metadata of USA Structures provides the following description of OCC_CLS: "This attribution identifies the top-level building occupancy class as defined by Locations: Building Occupancy Classification; FEMA Data Standard; July 31, 2018" [16]. The metadata also provides this description of PRIM_OCC: "This attribution identifies the primary descriptor for a building's usage for each top level building . . ." Overall, there are 10 OCC_CLS attribute domains that are further partitioned into 49 domain values within PRIM_OCC. Table 1 lays out the relationship of these two attributes and their domains. The following sections describe the conflation process and overall strategy ORNL uses to generate an occupancy classification for each structure in USA Structures dataset.

Table 1. OCC_CLS and PRIM_OCC domains

OCC_CLS	PRIM_OCC
Residential	Single Family Dwelling, Mobile Home, Multifamily Dwelling*, Temporary Lodging, Institutional Dormitory, Nursing Home
Commercial	Retail Trade, Wholesale Trade, Personal and Repair Services, Professional/Technical Services, Banks, Hospitals, Medical Office/Clinic, Entertainment/Recreation, Theaters, Parking, Veterinary/Pet
Industrial	Heavy, Light, Food/Drugs/Chemicals. Metal/Minerals Processing, High Technology, Construction
Agriculture	No Sub-classes
Assembly	Arena, Stadium, Convention Center, Religious, Social
Non-Profit	General Offices, Emergency Operation Centers
Government	General Services, Military, Emergency Services
Education	Pre-K - 12 Schools, Colleges/Universities, Libraries, Museums
Utility and Miscellaneous	Aviation, Ground, Marine, Rail, Power, Water Treatment

* denotes multiple subcategories

2.1 OCCUPANCY CLASSIFICATION AND CONFLATION

To be able to maximize the overall coverage of the occupancy attributes, the occupancy workflow incorporates a multitude of authoritative data sources, including 57 Homeland Infrastructure Foundation Level Data (HIFLD) data layers, Lightbox smart parcels, US Census housing unit data, Department of Housing and Urban Development (HUD), and Federal Aviation Administration (FAA) layers. The first three sources were used to determine the vast majority of structures' occupancy attribution. For some geographies, namely the Northern Mariana Islands and Puerto Rico, we also obtained local parcel coverage. Lastly, a binary classification model, named ResType, was developed and used to fill in data gaps where no coverage of the aforementioned sources exists.

With the complexity brought from the large number of various sources, both point and polygon feature representations, and the heterogeneous nature of the polygonal structure representations (e.g., a single polygon presentation can often encompass an entire row of buildings in a dense urban area), we elected to leverage a modeling framework described by Moehl [19] that allows for the precise modeling of the myriad data relationships. By splitting all of the structure polygons by all of polygons of the input data and then joining back attributions from those sources, we can precisely describe precisely the counts and amounts of overlaps and intersections among the sources and structures. This enables a rich description of the interactions among the various data sources, such as when one source labels a structure as a hospital but another classifies it as a nursing home. This highlights a limitation in our overall framework, which seeks to choose a single occupancy for each structure when some structures can equally be described by either label.

The order of the prioritized data layers considered in the overall occupancy classification is HIFLD, Lightbox smart parcels, and Census housing units. In general, if a structure intersects with HIFLD layer, then the structure occupancy will be determined by the type or theme of the HIFLD layer. If no intersection occurs, the next data source is used, which is the LightBox smart parcels. If the structure falls within a parcel that has a land use type appropriate for the USA Structure schema, then the parcel is used to determine the occupancy. If no occupancy has been determined by this step, the next source used is the Census housing unit data. Finally, if a structure remains unclassified, the ResType model evaluation, which exploits the building morphology of the structure, provides a final determination of the structures occupancy. There are exceptions to this overall process that will be covered in more detail in later sections. Figure 1 lays out the overall process.

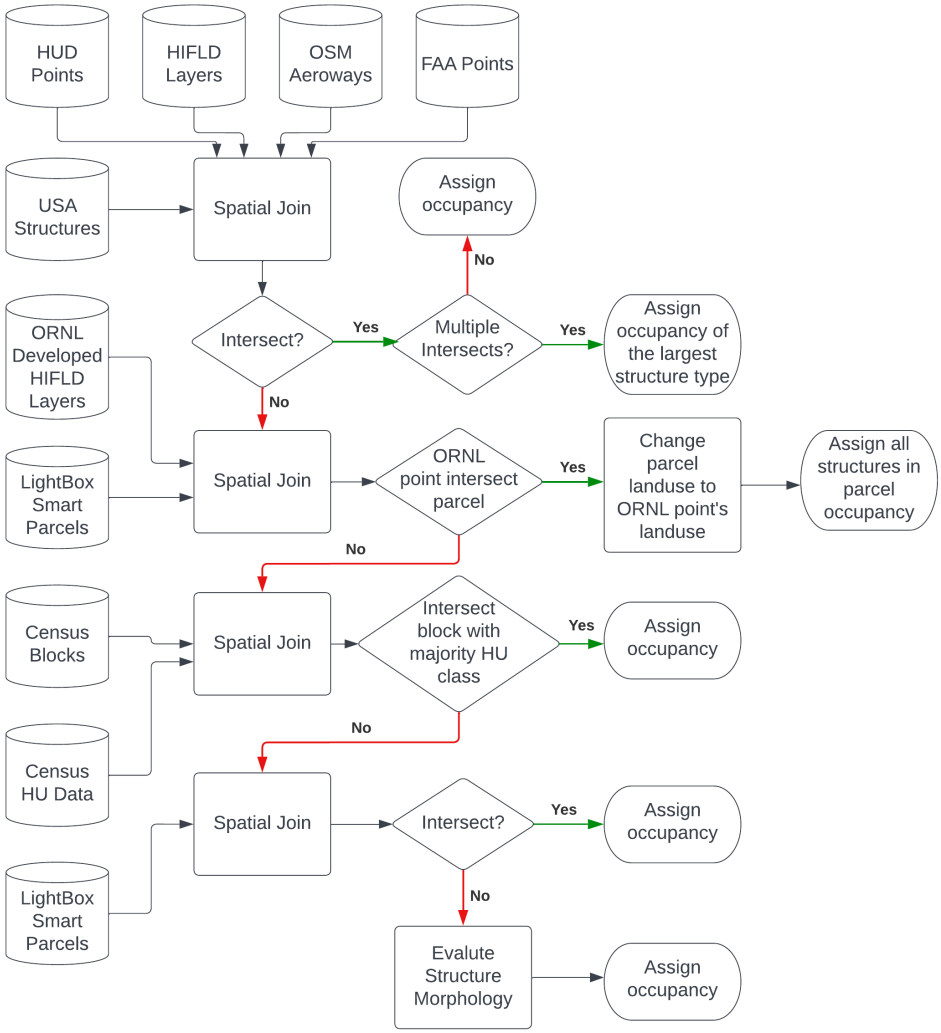

Figure 1. Occupancy Classification Workflow

2.2 DATA SOURCE EXTRACT TRANSFORM AND LOAD AND CONFLATION

The following sources are listed in order of preference for determining building occupancy. We have a brief description and notes about the implementation. For full descriptions, please see the source metadata where available.

2.2.1 Department of Housing and Urban Development

We are using two point datasets from the US Department of Housing and Urban Development (HUD) open data platform. These points represent addresses of properties that are assisted or insured through HUD [31], [32]. We currently use all 23,000 assisted and 17,000 insured property locations for analysis; however, in the future, we could filter based on geolocation accuracy. We are using this to designate structures within an intersecting parcel as “Multi-Family Dwelling” in the PRIM_OCC attribute. These

layers were acquired using the provided application programming interface (API) and loaded into our PostgreSQL database using Python and GeoPandas.

2.2.2 Federal Aviation Administration

We are using a polygon dataset that designates airport runways from the US FAA's open data platform. These polygons represent takeoff and landing areas [1]. Attribution is available for runway composition and length. For our analysis, we limit the 23,000 available records to a subset of 6,756 records by limiting to compositions of asphalt and/or concrete with a minimum length of 1,000 ft. We use the resulting subset to select intersecting Lightbox smart parcels that are then classified as "Aviation" in the PRIM_OCC attribute. These layers were downloaded from the open platform and loaded using Python and GeoPandas.

2.2.3 OpenStreetMap

We are using a polygon dataset from OpenStreetMap (OSM) that is a selection of all the polygons with the key "aeroway." This key is used for many features relating to airport structures [21], [20]. We use these polygons to select Lightbox smart parcels that are fully within, classifying them as "Aviation" in the PRIM_OCC attribute. We restored an OSM Planet file from February 25th using osm2pgsql to load into a PostgresSQL database and created a subset of features with the key "aeroway" for our osm_aeroways layer.

2.2.4 HIFLD

HIFLD is a foundational dataset related to domestic national security and emergency response. This collection of national geospatial datasets focus on mapping the nation's critical infrastructure and include standardization of schemas and attribution.

The HIFLD Extract Transform and Load (ETL) workflow includes the following steps:

1. Pull the data from the HIFLD API—Step one of the HIFLD ETL workflow is performed by a single Python script. The script calls the HIFLD REST API using the links provided in the data catalog on the HIFLD website. This script first reads the headers of the REST API to determine the maximum record pull. Each layer is pulled within partitions of the maximum record pull allowed. Each pull is received as a geoJSON object that is converted into a GeoPandas DataFrame. Code listing 1 highlights the procedure.

2. Ingest raw data into the PostgreSQL database—Each DataFrame from an API request is concatenated into a single data frame, projected to WGS 84, and ingested into a PostgreSQL database.

3. Transform all layers into a single production layer—Step three takes advantage of the PostgreSQL geometry data type, which can hold any geometry type in one field. This allows the users to call a single table for all HIFLD sources of each geometry type, reducing the complexity of the overall data integration process.

Table 2 lays out this crosswalk table mapping HIFLD layers to the USA Structure schema. This mapping was conducted by team members who heuristically mapped each layer to the most appropriate classification in the schema. Team members first matched each individually and then discussed where there was disagreement.

Table 2. Mapping HIFLD to PRIM_OCC and OCC_CLS

HIFLD Layer Name	OCC_CLS	PRIM_OCC
Agricultural Minerals Operations	Industrial	Food/Drugs/Chemicals
All Places of Worship	Assembly	Religious
Bio Diesel Plants	Industrial	Food/Drugs/Chemicals
Child Care Centers	Education	Other Educational Buildings
Colleges and Universities	Education	Colleges/Universities
Colleges and Universities campuses	Education	Colleges/Universities
Convention Centers and State Fairgrounds	Assembly	Convention Center
Courthouses	Government	General Services
DOD Sites Boundaries Public	Government	Non-Civilian Structures
DOD Sites Points Public	Government	Non-Civilian Structures
EPA Emergency Response Facilities	Government	General Services
Ethanol plants	Industrial	Food/Drugs/Chemicals
Ethanol trans loading facilities	Commercial	Wholesale Trade
FDIC Insured Banks	Commercial	Banks
Fedex Facilities	Commercial	Wholesale Trade
Ferrous Metal Mines	Industrial	Metals/Minerals Processing
Ferrous Metal Process Plants	Industrial	Metals/Minerals Processing
Fire Station	Government	Emergency Response
FDA Office Facilities	Government	General Services
Fortune 500 Corporate Headquarters	Commercial	Professional/Technical Services
General Manufacturing Facilities	Industrial	Light
Government Financial Processing Centers	Government	General Services
Governors Mansions	Government	General Services
Hospitals	Commercial	Hospital
Liquified Natural Gas Import Exports and Terminals	Commercial	Wholesale Trade
Local Emergency Operations Centers	Government	Emergency Response
Local Law Enforcement Locations	Government	Emergency Response
Major Sport Venues	Assembly	Indoor Arena
Major State Government Buildings	Government	General Services
Mines and Mineral Resources	Industrial	Metals/Minerals Processing
Miscellaneous Industrial Mineral Operations	Industrial	Metals/Minerals Processing
Natural Gas Processing Plants	Industrial	Food/Drugs/Chemicals
NCUA Insured Credit Unions	Commercial	Banks
Nonferrous Metal Mines	Industrial	Metals/Minerals Processing
Nonferrous Metal Processing Plants	Industrial	Metals/Minerals Processing
Nursing Homes	Residential	Nursing Home
Oil and Natural Gas Platforms	Industrial	Food/Drugs/Chemicals
Oil Refinery Polygon	Industrial	Food/Drugs/Chemicals
Oil Refineries	Industrial	Food/Drugs/Chemicals
Petroleum Ports	Commercial	Wholesale Trade
Petroleum Terminals	Commercial	Wholesale Trade
Pumping Stations	Commercial	Wholesale Trade
Power Plants	Utility and Misc	Energy Control Monitoring
Prison Boundaries	Residential	Institutional Dormitory

Private Non-Retail Shipping Facilities	Commercial	Wholesale Trade
Private Schools	Education	Pre-K - 12 Schools
Public Health Departments	Government	General Services
Public Refrigerated Warehouses	Commercial	Wholesale Trade
Public Schools	Education	Pre-K - 12 Schools
Sand and Gravel Operations	Industrial	Metals/Minerals Processing
Solid Waste Landfill Facilities	Utility and Misc	Ground
State Capitol Buildings	Government	General Services
Supplemental Colleges	Education	Colleges/Universities
Truck Driving Schools	Education	Other Educational Buildings
UPS Facilities	Government	General Services
Urgent Care Facilities	Commercial	Medical Office/Clinic
Veterans Health Administration Medical Facilities	Government	General Services

```
def getData(baseURL, tableName, schema):

    fields = "*"
    urlstring = f'{baseURL}?f=json'
    j = urllib.request.urlopen(urlstring)
    js = json.load(j)
    maxrcn = int(js["maxRecordCount"])
    where = "1=1"
    urlstring = f'{baseURL}/query?where={where}&returnIdsOnly=true&f=json'
    j = urllib.request.urlopen(urlstring)
    js = json.load(j)
    idfield = js["objectIdFieldName"]
    idlist = js["objectIds"]
    idlist.sort()
    numrec = len(idlist)
    fslist = []

    for i in range(0, numrec, maxrcn):

      torec = i + (maxrcn - 1)
      if torec > numrec:
        torec = numrec - 1
      fromid = idlist[i]
      toid = idlist[torec]
      where = "{} >= {} and {} <= {}".format(idfield, fromid, idfield, toid)

      urlstring = f'{baseURL}/query?where={where}&returnGeometry=true&outFields={fields}&f=geojson'
      resp = requests.get(urlstring, verify = False)
      data = resp.json()
      gdf = gpd.GeoDataFrame.from_features(data['features'])
      fslist.append(gdf)

    final_gdf = pd.concat(fslist)
    final_gdf= final_gdf.rename(columns=str.lower)

    final_gdf['layername'] = tableName
    final_gdf = final_gdf.set_crs(4326, allow_override=True)
    usadb.load_gdf_to_db(final_gdf, schema, tableName)
```

Listing 1. `getData()` function

2.2.5 PR Parcels

The parcel data used for Puerto Rico was developed by the Puerto Rico Planning Board. This dataset was published on November 5, 2021, and has 96 land use classifications that were aggregated specifically for the USA Structures schema. Google Translate was used, and those results were verified by bilingual staff at ORNL.

2.2.6 Lightbox Smart Parcels

The Lightbox smart parcels are provided through HIFLD licensed via a data agreement for federal use cases [8], [30]. The data are provided in several thousand layers and ESRI file geodatabases (per county and per res/nonres/unclassified combination). We use Python and OGR2OGR to load and append these into a single layer of Lightbox parcels. The data have more than 300 standardized land use attribute values and more than 180,000 nonstandardized land use attribute values. The standardized land use values were translated by FEMA, and a crosswalk table was provided to ORNL on September 29, 2021. This table describes how to aggregate specific land use codes into the the schema defined for USA Structures. Tables 3 and 4 describe the number of parcel land use codes that fall within each FEMA-defined domain.

Table 3. OCC_CLS landuse composition

OCC_CLS	Count of Landuses
Agriculture	30
Assembly	8
Commercial	102
Education	6
Government	24
Industrial	46
Residential	44
Unclassified	48
Utility and Misc	19

Table 4. PRIM_OCC landuse composition

PRIM_OCC	Count
Agriculture	30
Aviation	3
Colleges/Universities	2
Community Center	4
Construction	1
Emergency Response	1
Entertainment and Recreation	35
Food/Drugs/Chemicals	9
General Services	22
Ground	10
Heavy	4
High Technology	1
Hospital	2
Indoor Arena	2
Institutional Dormitory	4
Light	20
Manufactured Home	3
Marine	3
Medical Office/Clinic	4
Metals/Minerals Processing	11
Multi - Family Dwelling	16
Non-Civilian Structures	1
Nursing Home	1
Other Educational Buildings	1
Parking	3
Personal and Repair Services	8
Pre-K - 12 Schools	3
Professional/Technical Services	18
Rail	3
Religious	1
Retail Trade	26
Single Family Dwelling	12
Stadium	1
Temporary Lodging	8
Theaters	2
Unclassified	48
Veterinary/Pet	2
Wholesale Trade	2

2.2.7 US Census Bureau

The US Census Bureau provided a special tabulation of housing unit percentages at the block level from the 2010 census. These data are joined to the Tiger Census Block shapefiles. The dataset included five fields:

- **geoid**: Block level identification number
- **percent_sfr**: The percent of households within a block as single family residential single family residential
- **percent_mfr**: The percent of households within a block as multifamily residential
- **percent_mh**: The percent of households within a block as manufactured homes
- **percent_unassigned**: The percent of households within a block as unassigned buildings

2.3 SPATIAL CONFLATION

The vector analytical framework described in Figure 2.1 allows us to precisely define spatial relationships among the structure footprints and the attribution sources. We can also walk between and among indirect spatial relationships, such as from point to parcel to structure, to apply attribution. Each of these mechanisms and their applicable data sources will be described below.

2.3.1 Selection by Piece

The vector framework results in a table of polygon fragments with attribution for each input attribution layer. Sometimes, a structure will have more than one parcel polygon intersecting all or one of the vector framework fractions. For structures with multiple parcels that overlap a fragment of the structure, we select the land use of the parcel with the smallest area based on the assumption that a smaller parcel is more specific to the intersecting structure. This results in a table holding a record for each structure where multiple land uses are present with the land use of the smallest parcel denoted. We use this method for Lightbox and the Puerto Rico parcel datasets.

This process is represented in the Figure 1 flowchart by the **Intersect?**, **Multiple Intersects?** diamonds, the **LightBox Smart Parcels** and **Spatial Join** box.

2.3.2 Selection by Structure

For HIFLD points and polygons and parcel polygons, we construct tables denoting a land use for each structure Universally Unique IDentifier (UUID). The HIFLD data are often co-located on a single structure. To pick among the options, we first remove structures with only one point or polygon feature from HIFLD, HUD, and FAA, and assign an occupancy. Then, for structures with more than one point, we implement the logic that the point with the larger structure type will be preferred. To achieve this goal, we first calculate the average area for each occupancy class and sort the resulting table from largest to smallest area. We then assign an occupancy class with the largest area to a given structure.

This process is represented in the Figure 1 flowchart by the **Multiple Intersects?** diamond.

2.3.3 Selection by Parcel

In some situations, the direct intersection of source data layer with a structure is insufficient. For FAA runways, OSM aeroways, HUD points, and the HIFLD child care centers, colleges and universities, nursing homes, private schools, public schools, and supplemental colleges point layers, we calculate when a feature intersects or contains a parcel in the Lightbox smart parcel layer. We first generate a lookup for each parcel ID and source layer combination, and then we assign that occupancy class to the structure using the parcel ID. For accelerated intersecting operations, we employ a preprocessed parcel dataset in which geometries have been split to contain no more than 32 vertices. This allows for much faster polygon-to-polygon relation calculations.

This process is represented in the Figure 1 flowchart by the **ORNL point intersect parcel** diamond.

2.3.4 Selection by Census Block

To select Census housing unit data by Census block, we create a lookup for Census blocks based on a special tabulation. For each 2010 Census block, the tabulation has the percentage of housing units that are single family, multifamily, mobile home, or unassigned. We assign the entire block to a category if at least 95% of the units are any one of these types and the remaining units are unclassified. We then assign any structure in that block that is labeled "multifamily" as multifamily. Because there are often single structures such as churches or schools in a Census block that is otherwise single family, we attempt to prevent these structures being missclassified as residential by first calculating the mean size of a structure for each block and then calculating the standard deviation of these means, which is used to establish a large area threshold: the average mean plus two standard deviations. Structures that are under this area threshold in blocks that are classified as single family are then assigned to single family.

This process is represented in the Figure 1 flowchart by the **Intersect block with majority HU class** diamond.

2.4 GAUNTLET

GAUNTLET is a tool that has been developed alongside the USA Structures dataset since 2016. This tool generates building morphologies for each structure within the USA Structure dataset. GAUNTLET was originally designed to identify false positives in the raw building detection output from convolutional neural networks. The premise was that a false positive would have inherently different building morphologies than a true positive building object. We further expand the use of GAUNTLET in other scenarios that building morphologies are critical. A relevant use case for the GAUNTLET feature set was to model the difference between residential and nonresidential structures. We named a machine learning model that leverages GAUNTLET derived features as ResType model, which will be discussed in a later section.

The development of GAUNTLET was motivated by three main priorities: (1) finding and encoding useful features, (2) calculating those features as fast as possible, (3) and storing those features in an accessible and efficient manner. Table 5 shows the technical progress of GAUNTLET made to address these considerations.

2.4.1 Features

The 25 features that GAUNTLET generates fall into three main categories: geometric, engineered, and contextual measurements. Geometric measures are the common measurements of a geometry. For

Table 5. Gauntlet Technical Progression

Year	Features	Records/hour	Environment
2016	10	50,000	python/esri
2017	13	200,000	python/esri
2018	13	500,000	python/esri
2019	13	1,000,000	python/esri
2020-21	25	8,000,000	python/esri
2022	25	24,000,000	python/docker
2023	25	66,000,000	python/docker

example, area, perimeter length, and the vertex count are all geometric measures. Engineered measures are more sophisticated measurements based on geometric measurements. Complexity ratio and compactness index are two examples of engineered features. The last category of features are contextual measures, these measures come from scale analysis (SA) [4] and spatial point pattern analysis (SPPA) [7, 14] and describe a structure's relationship to its neighbors. We describe details of these three categories of features in the following subsections. Table 6 provides a brief descriptions of the morphology features generated by GAUNTLET.

2.4.2 Geometric Features

The four geometric features are area, perimeter, vertex count, and geom count. Geom count is the count of the number of geometries that are used to describe the structure detection. An example of this would be a structure detection with a courtyard within a structure, which causes a hole within the polygon. Two geometries, inner and outer, would be used to describe such a structure, so the geom count of this detection would be two.

2.4.3 Engineered Features

The next group of features is engineered features, which are vital for data exploration and often for machine learning success [9]. Feature engineering has proven successful in studies ranging from sentiment analysis [22] to text mining [6]. Among many possible engineered features in the literature, we selected several for consideration. The complexity ratio was first introduced by Ritter in 1822 [10, 15] and is the ratio of perimeter to the area. In 1978, Osserman proposed the IPQ [25], which is considered a more effective measure of compactness than the complexity ratio [15]. Despite the multitude of compactness measures proposed [2], Osserman's measure was used for its simplicity and low computational expense. IPQ is calculated in the following manner:

$$IPQ = \frac{4\pi}{P^2} \tag{1}$$

IPQ, also known as circularity [3], generates a value ranging from 0 to 1. The closer to 1 a geometry's IPQ is, the more circular the shape is. Certain shapes have specific IPQ values; for example, a perfect square has an IPQ of 0.785. Very inefficient shapes (i.e., shapes that do not maximize the area given a set perimeter length) typically have very low IPQ values. A geometry with a large hole or an L-shaped geometry are typical examples of these inefficient shapes. IPQ has been used in other fields to act as a measure of gerrymandering of voting districts [24]. Another engineered feature used is the Inverse Average

Table 6. Gauntlet Feature Set

Feature	Description
shape_area	Area of polygon in unprojected units
shape_length	Perimeter length in unprojected units
sqft	Area in square feet
lat_dif	The max latitude minus the min latitude in unprojected units
long_dif	The max longitude minus the min longitude in unprojected units
envel_area	The area of the bounding box of the geometry in unprojected units
vertex_count	The count of vertices in the geometry
geom_count	The count of polygons in the geometry
complexity_ratio	Shape_length/shape_area, a measure of how complex the shape is
iasl	Inverse average segment length
vpa	Vertices per area
complexity_ps	Complexity per segment, describes the average complexity within each segment
ipq	Isoperimetric quotient, describes how well a shape maximizes its area for the given perimeter
sqmeters	Area in square meters
n_count	Number of building centroids within 100 meters (min = 1 itself is included)
omd	Observed mean distance, the average distance of centroids within 100 meters
emd	Expected mean distance, the average distance if all centroids were uniformly spaced and equidistant
nnd	Distance between the centroid of a geometry to its nearest neighbor
nni	Nearest Neighbor Index, The overall pattern of points in the 100 meter buffer
intensity	The amount of nni occurring within the 100 meter buffer
n_size_mean	The average size of buildings within the 100 meter buffer
n_size_std	The standard deviation of building sizes within the 100 meter buffer
n_size_min	The smallest building size within the 100 meter buffer
n_size_max	The largest building sizes within with in the 100 meter buffer
n_size_cv	The Coefficient of variation of building sizes with in the 100 meter buffer

Segment Length (IASL), which is the vertex count divided by perimeter. The last engineered feature is the average complexity per segment, which is the complexity ratio divided by the vertex count.

2.4.4 Contextual Features

Contextual features are measures of various spatial relationships between the detection and the surrounding detections. All of these measures take place within a scan window that is centered at the detection's centroid. This scan window does not move across the detections at set increments but instead is centered on each detection before generating the following measures described below. The scan window is set as a fixed 100 meter radius circular buffer. There are two sets of contextual features: point pattern features, which measures spatial relationship of structure centroids; detection scale features, which ; and another that measures sizes of the detections within the scan window. Many of these features are well documented [4, 3] and have been used successfully in other classification problems [17, 13, 14].

2.4.4.1 Point Pattern Features

The neighborhood count is the count of all the structure centroids within the scan window. The nearest neighbor distance (NND) is the measured distance from one centroid of a detection the nearest neighboring detection's centroid. Both of these features are calculated using cKDtree from SciPy [36], which was

originally proposed by Maneewongvatana and Mount [18], and are used to derive more complex measures of spatial relationships [5].

The observed mean distance (OMD) is the sum of the nearest neighbor distances of all centroids within the scan window divided by the population of scan window [5]. The equation is as follows:

$$OMD = \frac{\sum d_i}{n} \tag{2}$$

EMD is the average distance of the closest centroid pairs if the centroids were at complete spatial randomness within the scan window [5]. Expected mean distance is calculated using the following equation:

$$EMD = \frac{1}{2 * \sqrt{\rho}} \tag{3}$$

The Nearest Neighbor Index (NNI) is OMD/EMD and has a range of 0–2.1491 [5]. In NNI, 0 represents a highly clustered pattern within the scan window, 1 represents a random distribution, and 2 or greater represents a pattern close to even dispersion within the scan window where all distances are close to equal [5]. The following equation presents the full formula for NNI:

$$NNI = \frac{\frac{\sum d_i}{n}}{0.5 * \sqrt{\rho}} \tag{4}$$

Intensity is the measure of how much NNI is happening within the window. For example, if two windows have the same NNI of 0.07, both windows have highly clustered events. However, if one of the windows has a higher intensity, more clusters or more members in the clusters are present in one scan window. Intensity is calculated using distances from the sample point (i.e., center of the scan window) to each individual point in the scan window [7]. The KDtree demonstrates its usefulness once again as we leveraged it to generate this metric. The formula for intensity is described in the equation below.

$$Intensity = \frac{\pi * \sum d_i^2}{n} \tag{5}$$

2.4.4.2 Detection Scale Features

The detection scale features are summary statistics of the detection sizes that are within the scan window. Most are straightforward and will be listed here. The smallest (NSmin) and largest (NSmax) structure sizes within the scan window are recorded as features for the structure whose centriod is being centered on by the scan window. Additionally, the mean (NSm) and standard deviation (NSs) of structure sizes within the window are captured the same way.

The coefficient of variation is the ratio of standard deviation to the mean. In this case, the coefficient is NSs/NSm and is abbreviated NScv. This number describes the homogeneity of detection sizes within the window. An NScv closer to 0 describes a scan window with similarly sized detections. Typically, an NScv of 1.3 or higher describes the presence of a single large detection surrounded by many smaller detections within the scan window.

2.5 RESTYPE MODEL

The ResType model is a supervised machine learning binary classification model. The purpose of this decision tree model is to fill data gaps for occupancy when a structure has no source information to provide a structure use. These gaps occur in the following situations:

- When a structure does not intersect a parcel
- When a parcel has no land use classification
- When a parcel has no land use classification that fits within the OCC_CLS or PRIM_OCC domains as defined by FEMA
- For structures labeled by Census Block type, if a structure was more than two standard deviations of the mean size, then the Census Block label is ignored
- We can optionally ignore a source classification of Unclassified and instead rely on ResType

The model's output is a residential probability that ranges from 0 to 100 per structure. The closer the inference is to either extreme, the more agreement the model's ensemble of decision trees has in its classification. The threshold we used in this work is 50. That is, the probability provided by ResType model greater than 50 is considered residential, but if the probability is less than 50, it is considered nonresidential. Since nonresidential does not fit within the schema of USA Structures, those structures will have an OCC_CLS domain value of "Unclassified."

2.5.1 Creation of Labels

To create training labels for binary classification ResType models, the parcel land use codes were aggregated into two categories: "residential" and "nonresidential." The structure geometries were spatially joined to the parcels. Structures that intersected more than one parcel had the area of each partial structure piece measured. The structure was then labeled according to the parcel with the largest coverage of the structure area. If the structure was joined to parcels with an equal area of coverage and those parcels had different land use classifications, they were excluded from the training set. This occurs when the parcels are duplicates. These aggregated land uses for each structure were added to the corresponding GAUNTLET feature set.

2.5.2 Detecting Bad Labels

We hypothesize that by removing anomalous records from our training datasets, we can expect to observe more accurate performance in subsequent classification models. Therefore, we developed an anomaly detection approach to identify less informative labels.

The rationale behind this approach is twofold. First, we attempt to aggregate all building use types into two broad categories: residential and nonresidential. At this high level of classification schema, there are structure instances that do not reasonably fit into the assigned labels. These instances can adversely influence the accuracy of the models to classify structures. Second, the provided labeled data is an engineered dataset, which is subject to curation by other parties and varying source sampling techniques. As such, some degree of incremental error creep is assumed as we receive the data for building ResType modeling.

A controlled experiment was conducted to determine a suitable anomaly detection treatment for the data. During the experiment, three unsupervised learning algorithms—Local Outlier Factor (LOF), Isolation Forest (ISO), and Stochastic Gradient Descent One-Class Support Vector Machine (SGDOSVM)—were applied to the data. The resulting inlier data identified were used to create training datasets for a baseline decision tree classifier. The performance of these three experimentally trained classifiers was compared against a classifier trained on untreated data. If a classifier subjected to treatment outperforms the control classifier, it can be inferred that the treatment received is the most effective at removing anomalous records from the data.

As shown in Table 7, the SGDOSVM approach outperformed the other three experimental classifiers, including the control classifier. As such, the feature dataset filtered on inliers from the SGDOSVM was used to create a training dataset for our robust occupancy type model. The output from the SGDOSVM results in two new attributes to filter by anomaly label and anomaly score. The anomaly label is binary and indicates if the data record is an inlier or outlier. The anomaly score provides a confidence value, where the higher the absolute value of the score, the greater the confidence in the anomalous nature of any given data record.

A key factor in anomaly detection is determining an optimal contamination rate for the ISO and LOF algorithms. For SGDOSVM, this parameter is known as ν. The most significant improvement in model performance was observed when setting the contamination rate between 32% and 35%, depending on the given sample. Note, anomaly detection was conducted on each individual binary class, implying that unique contamination rates per class should be determined and specified for future work.

A ν value of 0.35 was used for both classes. Future work should include determining the optimal ν value for each occupancy type class and work toward reducing the total amount of data being determined as anomalous.

Table 7. Accuracy Reports for Baseline Classifiers Assessing the Effects of Anomaly Detection

	Precision		Recall		F1-Score	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Control	0.70	0.77	0.80	0.66	0.75	0.71
ISO	0.72	0.86	0.89	0.66	0.80	0.74
LOF	0.69	0.80	0.85	0.62	0.76	0.70
SGDOSVM	0.91	0.94	0.94	0.91	0.92	0.92

	Accuracy		Training	Testing
	Macro Avg	Weighted Avg	Score	Score
Control	0.73	0.73	0.7342	0.7290
ISO	0.77	0.78	0.7793	0.7750
LOF	0.73	0.73	0.7410	0.7338
SGDOSVM	0.92	0.92	0.9273	0.9233

2.5.3 Training and Sampling

The ResType model consists of a submodel for each FEMA region. Each submodel was trained on a balanced class composition of nonanomalous residential and nonresidential labeled structures. As an example, Table **??** describes the class composition for Region 1 training data after identifying the

anomalous samples from the label set. The **Anomalous** and **Count** columns indicates the number of samples for anomalous (YES) samples and the samples (NO) will be used for the ResType model training.

Table 8. Region 1 Training Data

Class	Anomalous	Count
Residential	Yes	1,602,074
Residential	No	2,443,319
Nonresidential	Yes	201,356
Nonresidential	No	275,662

To create balanced classes the largest class was randomly downsampled to the smallest class. For Region 1, we randomly down sampled 2,443,319 residential labels to 275,662 to match the total of nonresidential labels. A gradient-boosted model was trained on 70% of the data and then tested on the remaining 30% of the data. Table 9 show the performance (precision, recall, and F1-score) of the Region 1 ResType model on the test set. We used the macro average and weighted average to capture the metrics in class imbalances in the test set. The **Support** column shows the number of samples counted as **Residential** or **Nonresidential** in the final classification result.

Table 9. Region 1 Classification Report

Class	Precision	Recall	F1-score	Support
Residential	92	95	93	82624
Nonresidential	94	92	93	82662
Accuracy	N/A	N/A	93	165286
Macro Average	93	93	93	165286
Weighted Average	93	93	93	165286

2.6 LESSONS LEARNED AND FUTURE WORKS

- Anomaly detection of labels
  During the creation of the training data, we observed a spike in model accuracy performance near the identified ν value of 0.35. Assuming that 35% of our data is anomalous and should be removed is a naive assumption. Doing this would not only remove anomalous data records, but it would also remove a significant portion of complex building features. Consequently, this could make our model less generalizable. Therefore, future work should focus on reducing the ν value to a more conservative estimate in the aim of preserving complex building features and removing the highest likelihood anomalous instances. The controlled experiment approach is useful for observing the effects of anomaly detection methods and selection when curating a training dataset; however, the results can be misleading when using it as an approach for estimating a contamination rate or ν value. For future work, we should identify an estimate that allows for adequate coverage of possible anomalous records while minimizing the risk of removing truthful complex records.
- Developing regional ResType models
  The use of FEMA Regions extent to develop ResType models was our first attempt and a rather convenient choice. In the future, we might need to revisit this decision. For example, it likely that an island territory model that includes territories from different FEMA Regions will have better

performance rather than being a part of a model that included with a model where most of its training data comes from the states. Additionally, there is a need in the future to create the third class to capture to type of multifamily residential. For example, apartments are often quite different in morphology from single family residences and nonresidential buildings.

3. ADDRESS ASSIGNMENT

This update to USA Structures also includes address attribution, a key datum for the emergency preparedness and response community. After an event, for example, FEMA receives many requests for assistance from those affected by the disaster, but the validity of each request must be confirmed before relief funds can be granted. Including address information in the USA Structures dataset allows FEMA to more quickly and effectively search for the address listed in a relief request and verify the impact by the event. More generally, addresses are the most common means of identifying structures, so by including these information, USA Structures can be easily leveraged along with other datasets and thereby serve a wider variety of applications. However, accurately conflating structure polygons with open source address information presented numerous challenges. We outline our solutions to those issues below.

3.1 ADDRESS SOURCES AND ETL

The address data included in USA Structures were derived from publicly available, open source data. Although we identified some open sources published by individual states, the primary source for addresses was the National Address Database (NAD), a US Department of Transportation–led effort to collate and distribute a standardized geospatial dataset of addresses in the United States [26]. As of March 2023, the US Department of Transportation had partnered with state and local governments in 45 states to deliver address data covering most of the United States, though some partners have yet to provide data. In those areas without NAD coverage, we identified available state sources; however, some states either have no open address data or do not make them available to the public, so gaps in address information are present in some areas of USA Structures.

The source address data were loaded into our PostgreSQL database using a variety of Python functions that utilize the ogr2ogr package [11], depending on the source format.

3.2 ADDRESS PROCESSING AND RANKING

Although address data schema vary by source, most address data conforms to a similar structure that we accounted for in our address processing script. Before execution, we reviewed each source, identified the target fields, and tailored the script to capture the street address, city name, postal code, state, and geometry. In some sources, these data were stored in six fields, but in others, such as the NAD, the target data were segmented into numerous address components (e.g., address number prefix, address number, address number suffix, street name premodifier, street name predirectional, street name pretype, street name, street name post-type).

Once the target fields were identified, our processing script separated the address number from the address street name; alternatively, if already separated, as was the case with the NAD, the associated components of address number (e.g., address number prefix, address number, and address number suffix) and street name (e.g., street name premodifier, street name predirectional, street name pretype, street name, street name

post-type) were combined into two elements, respectively. The script then performed a series of cleaning steps and logical tests on each element.

Address number:

- Removed leading and trailing white space
- Removed invalid special characters (excluding "-" and "/," which are valid characters in some address numbers)
- Verified that the element contained a number

Street name:

- Removed leading and trailing white space
- Converted text to uppercase
- Verified that the name contained only alphanumeric characters
- Concatenated name element with number element and verified that the resulting string contained two or more valid elements (i.e., an address number and street name)

City name:

- Removed leading and trailing white space
- Converted text to uppercase
- Removed special characters
- Verified that the element contained no numeric characters
- Verified that the element contained three or more characters

Postal code:

- Removed leading and trailing white space
- Verified that the element contained only numeric characters
- Verified that the element contained five characters

Rather than using state information provided by the source directly, we performed a spatial join of all data with state geometry data from the US Census. This ensured that the state field was fully populated and standardized.

As an additional measure of quality control, we cross-referenced all city name, postal codes, and state pairings in the address source data with verified combinations of those data from authoritative sources including the US Postal Service, US Geologic Survey, US Census Bureau, open source data, and HERE geospatial data [34, 35, 33, 29, 28, 27, 23, 12]. Address elements that were not found in these reference tables were excluded from the final processed address table as a verification and validation step.

Finally, each record was assigned a rank based on the validity of its street address, city name, and postal code information, as determined by the criteria above. In the early stages of developing an address conflation workflow, in areas where address data from different sources overlapped, we identified the need

to assess the quality of each address record so as to make a more informed decision when selecting a final address and assigning it to a given structure. To that end, each record was assigned a rank of 10 and then for each valid address element—street address, city name, and postal code—the rank was improved. The amount of improvement was weighted differently by element, such that a valid street address improved a record's rank by 4, city name by 2, and postal code by 1. In this way, we could ensure that records with more specific address information were weighted more heavily in the final address conflation. Table 10 provides examples of address records and how they would be ranked according to our methodology. As shown in those examples, records with more complete and valid data are ranked better than records with less complete and valid data, and more complete records are more likely to be selected as the final address during address conflation, which is described in the next section.

Table 10. Address rankings with examples

Address Example	Rank
101 Smith Rd, Unit B, Pleasantville, VT, 05231	1
101 Smith Rd, Pleasantville, VT, 05231	2
101 Smith Rd, Pleasantville, 05231	3
101 Smith Rd, Pleasantville	4
101 Smith Rd, 05321	5
101 Smith Rd	6
Pleasantville, 05321	7
Pleasantville	8
05321	9
No valid data	10

3.3 ADDRESS CONFLATION

3.3.1 Priority Attributions

After the address datasets have been cleaned and ranked, we can then begin the process of associating them with structures. First, we have ranked the addresses to allow us to select the best address in terms of attribution when multiple options from within or among sources are available. We have also listed address sources in order of priority for each state. Texas and Florida have local address sources that are prioritized, otherwise we defer to NAD over OA. However, we prioritize first on the completeness of the address rank, then the source. For example, we have an address from both the National Address Database and Open Addresses, we would choose the more complete address from Open Addresses, even if this were farther away than the NAD point. Similarly, although we prefer state sources (i.e., Texas HAND address data in Texas and Florida Parcel Data Statewide in Florida), if a structure has a more complete address from NAD within the geolocation threshold, we will take the NAD address over the Texas HAND. If multiple points have the same rank, we defer to the higher priority source. Our code is designed to flexibly allow or exclude datasets depending on redistribution restrictions. This allows us to assess and report the impact of restrictions on the address attribution process.

3.3.2 Geospatial Linkages

We use the known characteristics of the address data to determine the best geolocation mapping for selection. Some address points are on an entity, or rooftop, therefore we can assume that if an address point intersects a structure, that address can be assigned to that structure. Intersection can also be used in the

opposite direction if the address source is polygonal, such as is the case with Florida's parcel dataset. If a structure centroid intersects a parcel, we assume the address can be assigned to this structure.

After assigning structure addressed based on intersections, we select the structures that did not get an address from intersection or that have a rank higher than 6. We then calculate the nearest addresses by intersecting the addresses and structures with parcels. A structure can only be assigned an address if it is within the same parcel and within 350 ft of the address point.

For structures that do not get an address from either of the above workflows, we then select addresses from Lightbox parcels where the structure intersects a parcel. These are done in a separate query so that selections can be limited when there is a redistribution restriction in place. This allows us to easily provide Lightbox parcel information for internal FEMA use and only non-Lightbox for external distribution.

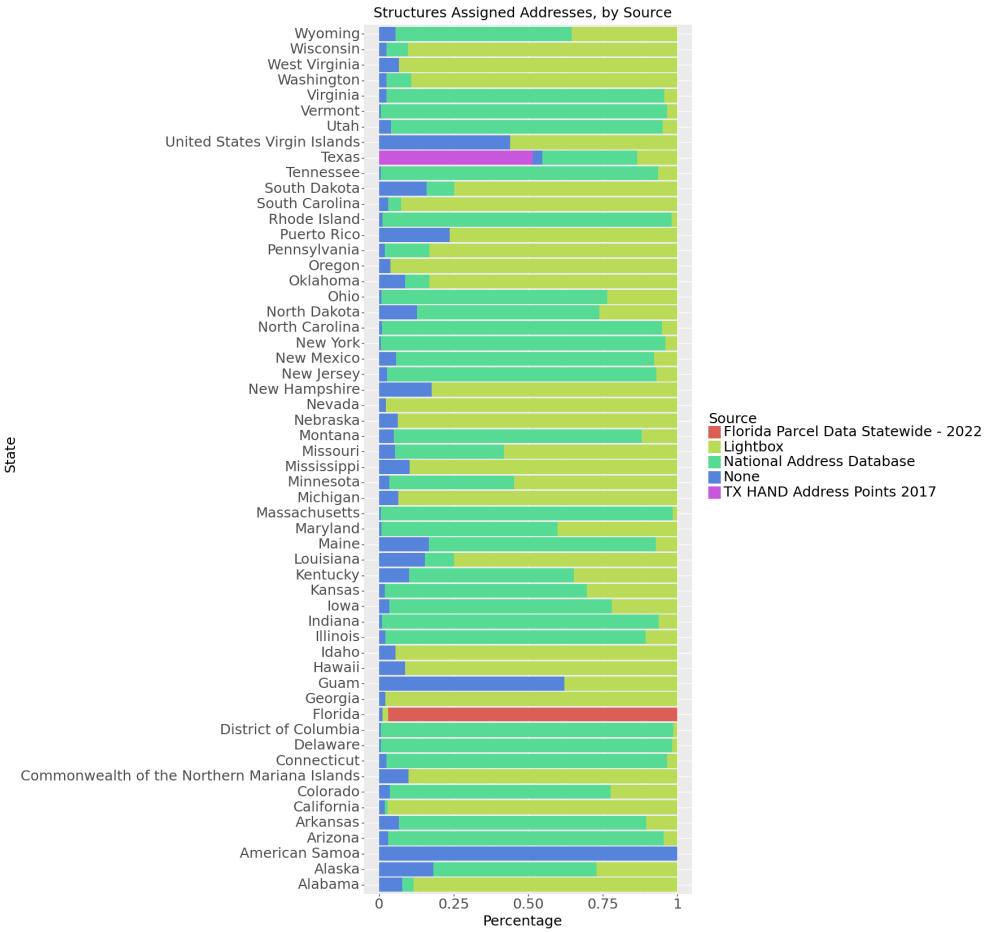

Figure 2. Percentage of addresses by source, by state.

3.4 LESSONS LEARNED AND FUTURE WORKS

Based on our testing and observations, we found that the methodology outlined above is a reliable means of conflating many valid addresses to our structure polygons. However, there are also some limitations and opportunities for improvement.

Firstly, missing addresses in our structure data often reflect gaps in the availability of open source address data. Secondly, the steps we took to perform QA/QC checks, rank address records based on validity and completeness, and leverage ancillary datasets to guide the conflation process cannot compensate for poor data quality. For example, imprecise geolocations, such as those derived from linear referencing along street network centerlines, as well as invalid address elements, resulted in poor address conflation results in some areas. Some of these issues could be mitigated through improvements to our methodology, but artifacts of these issues will be present in the data until the quality of the source data improves.

We would also like to mention that our methodology has some notable limitations. Firstly, it was primarily designed to process addresses that are typical for structures in the continental United States. However, addresses found in the US territories can be very different. While we took steps to tweak our address ranking process to account for some of these differences, further refinement is needed to more accurately capture addresses in those areas.

Likewise, our methodology currently does not account for multiaddress structures, such as townhouses, urban city blocks, and strip malls. Unlike many apartment buildings, which typically have a single street address with varying unit numbers, the aforementioned structures could have multiple street addresses with varying street numbers for a single contiguous structure. According to our approach, only one of those addresses would be captured for the structure. Future work would focus on a more comprehensive approach to account for multiaddress structures.

Finally, although our interpretation of the results suggest that many of the structure-address pairings are reliable, the overall accuracy of our conflation methodology is uncertain and is a priority for future work. Measuring accuracy presents numerous challenges. Most notably, we will need an authoritative address source with which to compare our results. While we consider the NAD to be the best available public source for address data, we have found some errors therein, which were inherited from their data participants. Ideally, these data would be obtained from the US Postal Service, but the availability of that data is unknown.

4. END USER PORTAL

A web-based repository has been created to allow sponsors to retrieve the latest data or reacquire previously released data. The web application leverages the content management system (CMS) Drupal, making it easy to publish data products without additional code. The Drupal CMS also allows for comprehensive user management to allow access to the website and provides download privileges. The FileBrowser extension was added to this deployment of Drupal to add file explorer–like functionality. The web application is patched regularly using Lando, a DevOps tool built on Docker's containerization platform. This allows for continuous integration and continuous deployment, which adds automation to the application development process

Users are added to the platform by a site administrator based on requests from the sponsor. A username and password is provided to the new end user in a series of two emails. User name and login information is sent to sponsors and ORNL project members. A password email is only sent to the requesting user. Deliverables are logically grouped to allow end users to quickly navigate the folder structures to locate the desired download. Figure 3 shows the grouping, and Figure 4 shows an example of folder name and description of the folders contents. Clicking on a folder will show the available downloads. Selecting a download option from the list will start the download process of a compressed deliverable. The naming

convention followed for deliverables is the text deliverable, the data of the deliverable (YYYYMMDD), and the 2 character state abbreviation (e.g., Deliverable20230331CT).

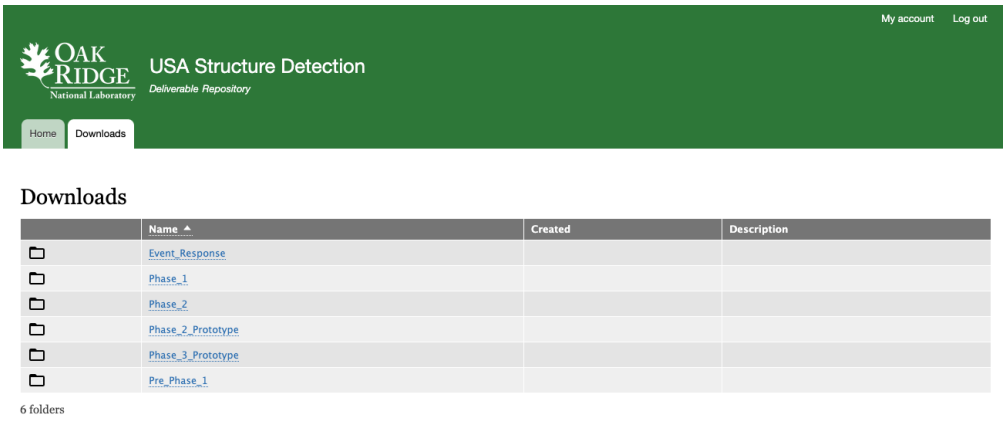

Figure 3. The overview of end user data portal

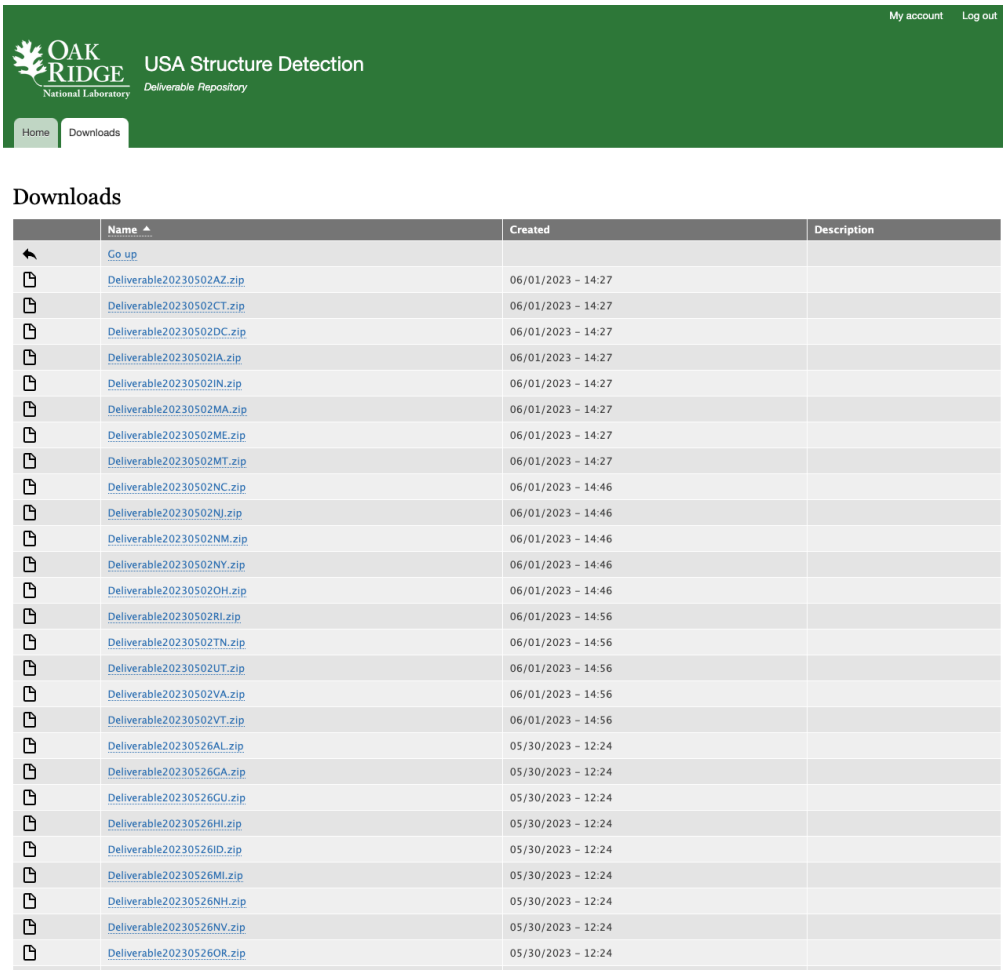

Figure 4. An example of filenames and folders for downloading.

References

[1] US Federal Aviation Administration. *Runways*. Data retrieved on 2-27-2023 from https://adds-faa.opendata.arcgis.com/datasets/faa::runways/about. 2023.

[2] Shlomo Angel, Jason Parent, and Daniel L. Civco. "Ten Compactness Properties of Circles: Measuring Shape in Geography". In: *Canadian Geographer* 54.4 (2010), pp. 441–461. DOI: https://doi.org/10.1111/j.1541-0064.2009.00304.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1541-0064.2009.00304.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0064.2009.00304.x.

[3] Melih Basaraner and Sinan Cetinkaya. "Performance of Shape Indices and Classification Schemes for Characterising Perceptual Shape Complexity of Building Footprints in GIS". In: *International Journal of Geographical Information Science* 31.10 (2017), pp. 1952–1977. DOI: 10.1080/13658816.2017.1346257. eprint: https://doi.org/10.1080/13658816.2017.1346257. URL: https://doi.org/10.1080/13658816.2017.1346257.

[4] Filip Biljecki and Yoong Shin Chow. "Global Building Morphology Indicators". In: *Computers, Environment and Urban Systems* 95 (2022), p. 101809. ISSN: 0198-9715. DOI:

https://doi.org/10.1016/j.compenvurbsys.2022.101809. URL: https://www.sciencedirect.com/science/article/pii/S0198971522000539.

[5] Philip J. Clark and Francis C. Evans. "Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations". In: *Ecology* 35.4 (1954), pp. 445–453. ISSN: 00129658, 19399170. URL: http://www.jstor.org/stable/1931034 (visited on 06/15/2022).

[6] Gordon V. Cormack, José Maria Gómez Hidalgo, and Enrique Puertas Sánz. "Feature Engineering for Mobile (SMS) Spam Filtering". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. Amsterdam, The Netherlands: ACM, 2007, pp. 871–872. ISBN: 978-1-59593-597-7. DOI: 10.1145/1277741.1277951. URL: http://doi.acm.org/10.1145/1277741.1277951.

[7] J.P. Diggle. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. Chapman and Hall/CRC, 2013. DOI: https://doi.org/10.1201/b15326.

[8] DMP/Lightbox. *SmartParcels*. Licensed through HIFLD Secure Data. July 2021. (Visited on 07/16/2021).

[9] Pedro Domingos. "A Few Useful Things to Know About Machine Learning". In: *Commun. ACM* 55.10 (Oct. 2012), pp. 78–87. ISSN: 0001-0782. DOI: 10.1145/2347736.2347755. URL: http://doi.acm.org/10.1145/2347736.2347755.

[10] Y.S. Frolov. "Measure Shape of Geographical Phenomena—History of Issues". In: *Soviet Geography Review and Translation* 16.10 (1975), pp. 676–687.

[11] GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. 2020. URL: https://gdal.org.

[12] HERE. *HSIP Gold (HERE) Geocoding Data*. City points of interest (POIs) and postal code boundaries were used as reference. 2020. URL: //ornl.gov/gistdata/DATA/HSIPGold/HSIP_GOLD_2020/HERE_2020/NorthAmerica.gdb.

[13] Warren Jochem, Tomas Bird, and Andrew Tatem. "Identifying residential neighbourhood types from settlement points in a machine learning approach". In: *Computers, Environment and Urban Systems* (Jan. 2018), pp. 1–10. URL: https://eprints.soton.ac.uk/417168/.

[14] Warren C. Jochem, Tomas J. Bird, and Andrew J. Tatem. "Identifying residential neighbourhood types from settlement points in a machine learning approach". In: *Computers, Environment and Urban Systems* 69 (2018), pp. 104–113. DOI: 10.1016/j.compenvurbsys.2018.01.004. URL: https://doi.org/10.1016/j.compenvurbsys.2018.01.004.

[15] Wenwen Li, Michael F. Goodchild, and Richard Church. "An Efficient Measure of Compactness for Two-Dimensional Shapes and its Application in Regionalization Problems". In: *International Journal of Geographical Information Science* 27.6 (2013), pp. 1227–1250. DOI: 10.1080/13658816.2012.752093. eprint: https://doi.org/10.1080/13658816.2012.752093. URL: https://doi.org/10.1080/13658816.2012.752093.

[16] *Locations: Building Occupancy Classifications*. Tech. rep. 500 C St SW Washington, DC: Federal Emergency Management Agency, Response Geospatial Office, 2018.

[17] Z. Lu et al. "Building Type Classification Using Spatial and Landscape Attributes Derived from LiDAR Remote Sensing Data". In: *Landscape and Urban Planning* 130 (2014), pp. 134–148.

[18] Maneewongvatana, Songrit and Mount, David M. *Analysis of Approximate Nearest Neighbor Searching with Clustered Point Sets*. 1999. DOI: 10.48550/ARXIV.CS/9901013. URL: https://arxiv.org/abs/cs/9901013.

[19] Jessica Moehl, Eric Weber, and Jacob McKee. "A Vector Analytical Framework for Population Modeling". In: *FOSS4G 2021*. Buenos Aires: ISPRS, Sept. 2021.

[20] OpenStreetMap Contributors. *Key:aeroway OpenStreetMap Wiki*. Accessed on 7-3-2023 https://wiki.openstreetmap.org/wiki/Key:aeroway. 2023.
[21] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.
[22] Bo Pang and Lillian Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. ACL '04. Barcelona, Spain: Association for Computational Linguistics, 2004. DOI: 10.3115/1218955.1218990. URL: https://doi.org/10.3115/1218955.1218990.
[23] Paul Ellis (pseudosavant). *USPS ZIP Codes Lookup*. Accessed on 5-8-2023. 2023. URL: https://github.com/pseudosavant/usps-zip-codes/blob/main/dist/ZIPCodes.json.
[24] D. Polsby and R. Popper. "The Third Criterion: Compactness as a Procedural Safeguard against Partisan Gerrymandering". In: *Yale Law & Policy Review* 9 (1991), pp. 301–353. URL: http://www.jstor.org/stable/40239359.
[25] Osserman R. "Isoperimetric inequality". In: *Bulletin of the American Mathematical Society* 84.6 (1978), pp. 1182–1238.
[26] US Department of Transportation. *National Address Database*. 2023. URL: https://www.transportation.gov/gis/national-address-database.
[27] US Census Bureau. *2020 Redistricting Data (P.L. 94-171) Name Lookup Tables (NLTs)*. Accessed on 5-8-2023 and downloaded all 50 states, the District of Columbia, and the Commonwealth of Puerto Rico. 2020. URL: https://www.census.gov/geographies/reference-files/time-series/geo/name-lookup-tables.html.
[28] US Census Bureau. *2020 TIGER/Line Shapefiles: ZIP Code Tabulation Areas*. Accessed on 5-3-2023. 2020. URL: https://www2.census.gov/geo/tiger/TIGER2020/ZCTA520/tl_2020_us_zcta520.zip.
[29] US Census Bureau. *2020 ZIP Code Tabulation Area (ZCTA) Relationship File*. Accessed on 5-8-2023. 2020. URL: https://www2.census.gov/geo/docs/maps-data/data/rel2020/zcta520/tab20_zcta520_county20_natl.txt.
[30] US Department of Homeland Security. $LightBox_{PARCEL}{}_{LICENSE}{}_{2}020.pdf$. accessed on 7-6-2023 at https://geoplatform.maps.arcgis.com/sharing/rest/content/items/17065b495f0a4e4f85a8509d554beea3/data. 2023. (Visited on 07/06/2023).
[31] US Department of Housing and Urban Development. *HUD Insured Multifamily Properties*. Data retrieved on 2-27-2023 from https://services.arcgis.com/VTyQ9soqVukalItT/ArcGIS/rest/services/HUD_Insured_Multifamily_Properties/FeatureServer, 2023.
[32] US Department of Housing and Urban Development. *Multifamily Properties—Assisted*. Data retrieved on 2-27-2023 from https://services.arcgis.com/VTyQ9soqVukalItT/arcgis/rest/services/Multifamily_Properties_Assisted/FeatureServer. 2023.
[33] US Geologic Survey. *US Geologic Survey National File*. Accessed on 5-8-2023. 2023. URL: https://geonames.usgs.gov/docs/stategaz/NationalFile.zip.
[34] US Postal Service. *US Postal Service Area and District File*. Accessed on 5-2-2023. 2023. URL: https://postalpro.usps.com/storages/2023-05/AREADIST_ZIP5.TXT.
[35] US Postal Service. *US Postal Service Locale Detail*. Accessed on 5-8-2023. 2023. URL: https://postalpro.usps.com/mnt/glusterfs/2023-05/ZIP_Locale_Detail.xls.
[36] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.