Portfolio

Prawan Amatya
Geographic Information System

Hurricane Harvey:
Property Damage and Housing Price Difference Assessment

Cartography and Visualization
Prawan Amatya

Introduction:

On August 25, 2017 through August 30, 2017, Hurricane Harvey's rains triggered the catastrophic flooding in south Central Texas, the greater Houston metropolitan area, and Southeast Texas. This project however, only focuses on the property damage caused by Harvey. Texas Department of Insurance (TDI) hosted Texas State Disaster Coalition conference calls for the Hurricane Harvey responses, with the insurance companies. The call issues a data call from the insurers to monitor the financial impact of Hurricane Harvey on the market in terms of claims handling, data feed for policy makers and public officials. The collected data for total number of claims and Total Loss in Dollars were according to ZIP code. After the data call, Hurricane Harvey generated about 670,000 claims (354,000 for residential property claims and 203,000 automobile claims) to private insurers, TWIA, and the Texas FAIR Plan for all personal and commercial lines of insurance. By the time of data reporting the insurers have made $4.5 billion in claim payments and estimates the overall payout total amount of $15.7 billion. Even though most of the claims were reported for residential property, automobile insurance is covered for flooding damage, while residential property does not cover for flood. Thus, fluctuation of single family housing closing price for the month of June in 2017 versus 2018 is compared caused by the impact of Hurricane Harvey triggered flood.

Study Area:

Most of the damage in Texas after the flooding triggered by Hurricane Harvey happened in south Central Texas, the greater Houston metropolitan area, and Southeast Texas. Keeping in mind the scope of project, greater Houston metropolitan area is selected. ZIP code areas are used in this project to visualize the data as the provided data of the claims and incurred loss were collected in terms of ZIP code by Texas Department of Insurance (TDI).

Methods:

A Bivariate map is created for the TDI dataset for number of claims and incurred loss per Zip code

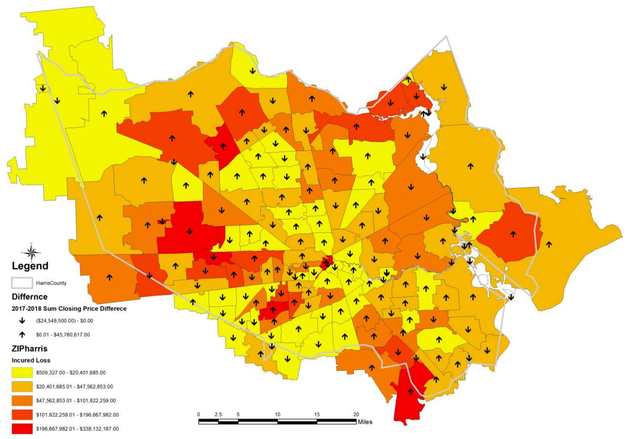
Dataset for the closing price of Single Family Housing units for the Month of June 2017 and June 2018 from Harris county is geocoded and created a sum spatial joint with Zip Code shapefile.

A bivariate map is created for the summed closing price per zip code for the Month of Jun 2017 and June 2018.

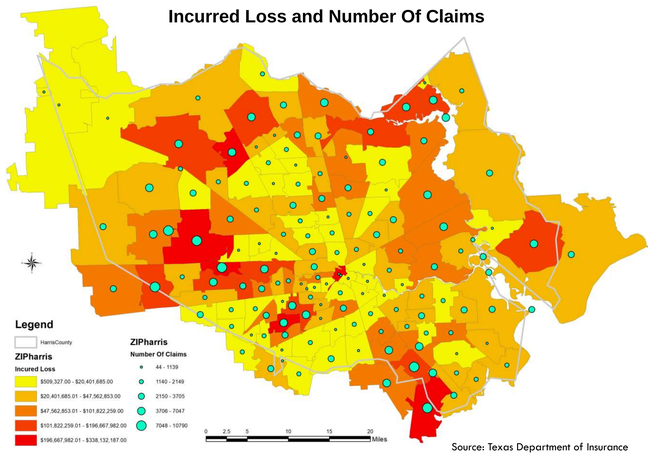
Another bivariate map is created to compare the Incurred Loss versus the Price Difference between the Summed closing price of the Month of Jun 2017 and Jun 2018.

Finally, a multi-Variate map is created to compare the data amongst the Bayou, floodway and Summed closing price difference of Jun 2017 and Jun 2018

Incurred Loss and Housing Closing Price Difference

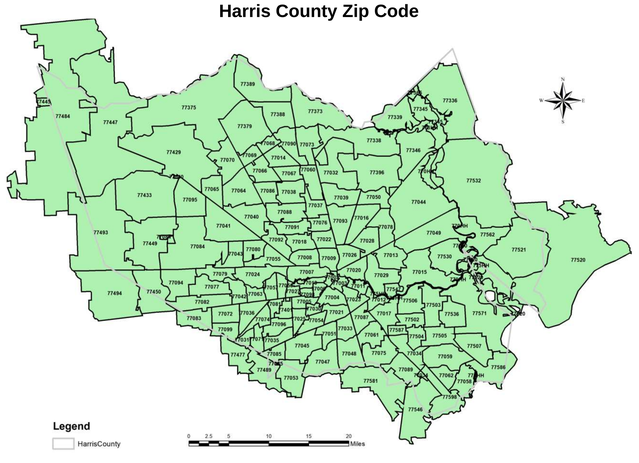


Incurred Loss and Number Of Claims

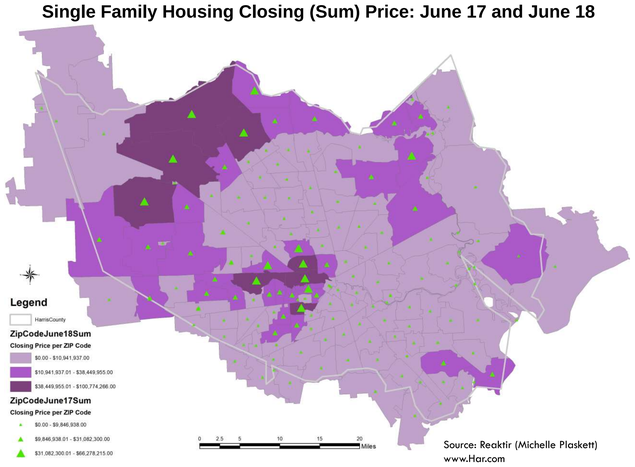


Source: Texas Department of Insurance

Harris County Zip Code

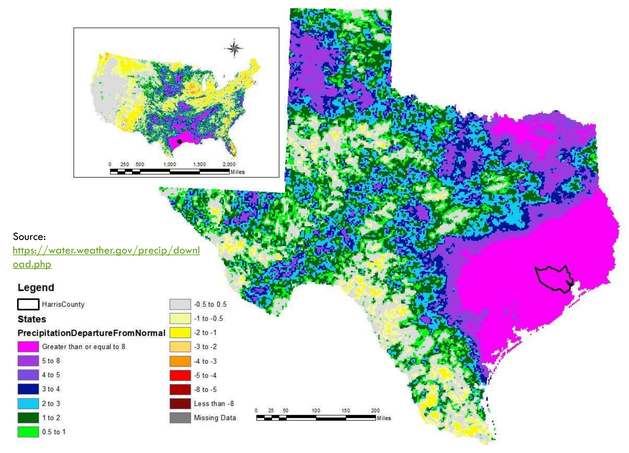


Single Family Housing Closing (Sum) Price: June 17 and June 18

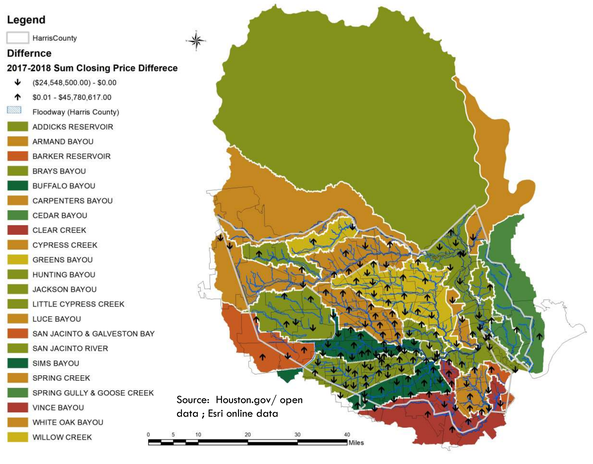


Source: Reaktir (Michelle Plaskett) www.Har.com

Precipitation : August 2017



Watershed, Floodways and Housing Closing Price Difference



Observation / analysis

There seemed to be direct correlation between Texas Department of Insurance (TDI) number of claims versus incurred loss in the designated zip code. The Closing price difference of the Single Family Housing from June 2017 and June 2018 is compared to the Incurred Loss for the individual Zip Code indicated somewhat relevance. The Incurred Loss value not only depended on the number of claims, it also depended on the individual housing sold price, vehicles and all the other insurable losses in the zipcode. Finally, the price difference was compared against the Bayou and Floodway, the relevance is seen where the majority of Zip Code was influenced by the floodway and larger Bayou.

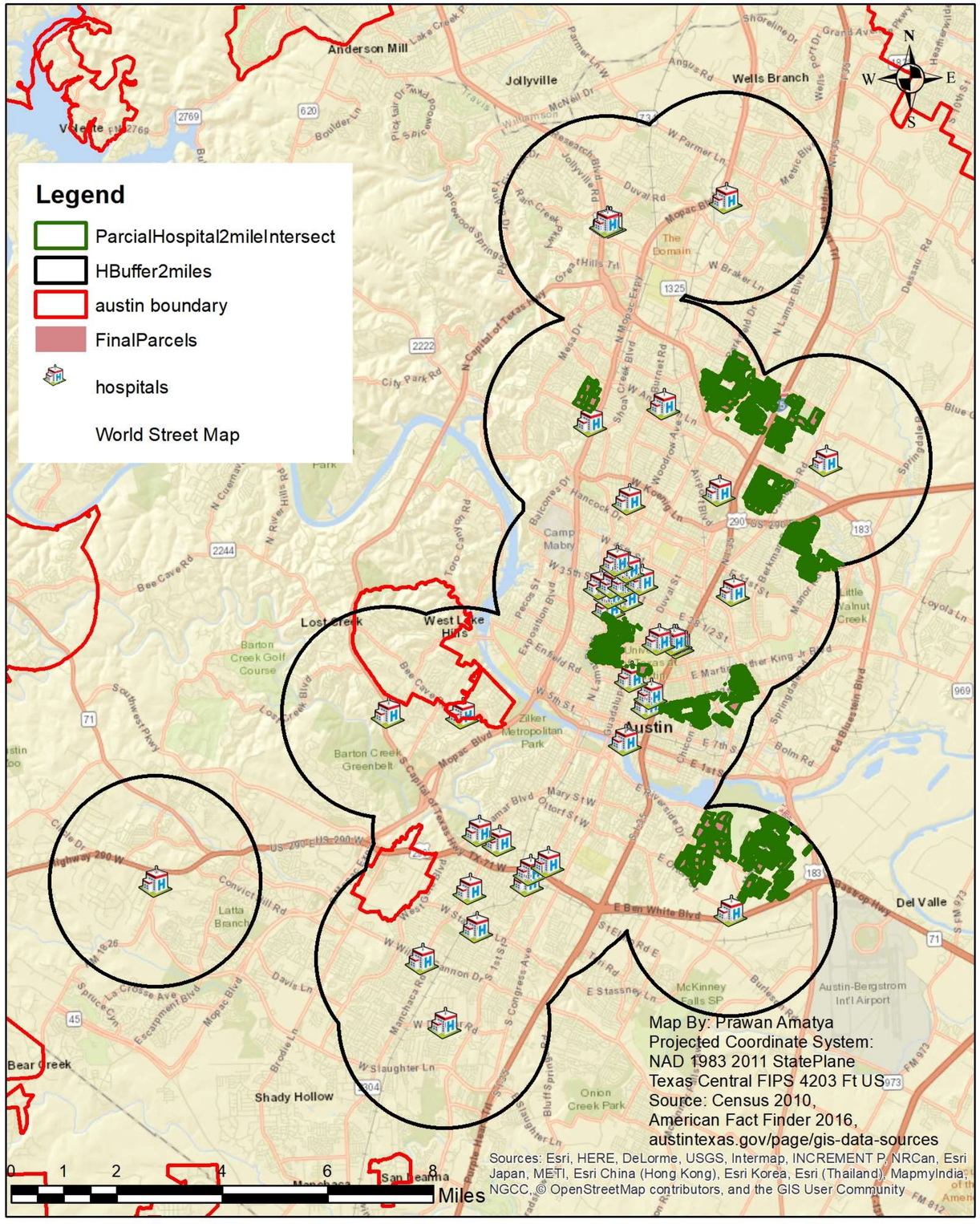
Result/ Conclusion

The Bi-variate and multi-variate map proved to be very useful in comparing the data to see the correlation amongst various geographic data and events. As for the data result, the housing price difference seems to only corelate in the certain zip codes. There are some relevance to the price change and Floodways, but the rise and fall of the Housing price also depends on other factors like, the Economy, Interest Rates, Area Desirability, Political Forces and Supply and Demand.

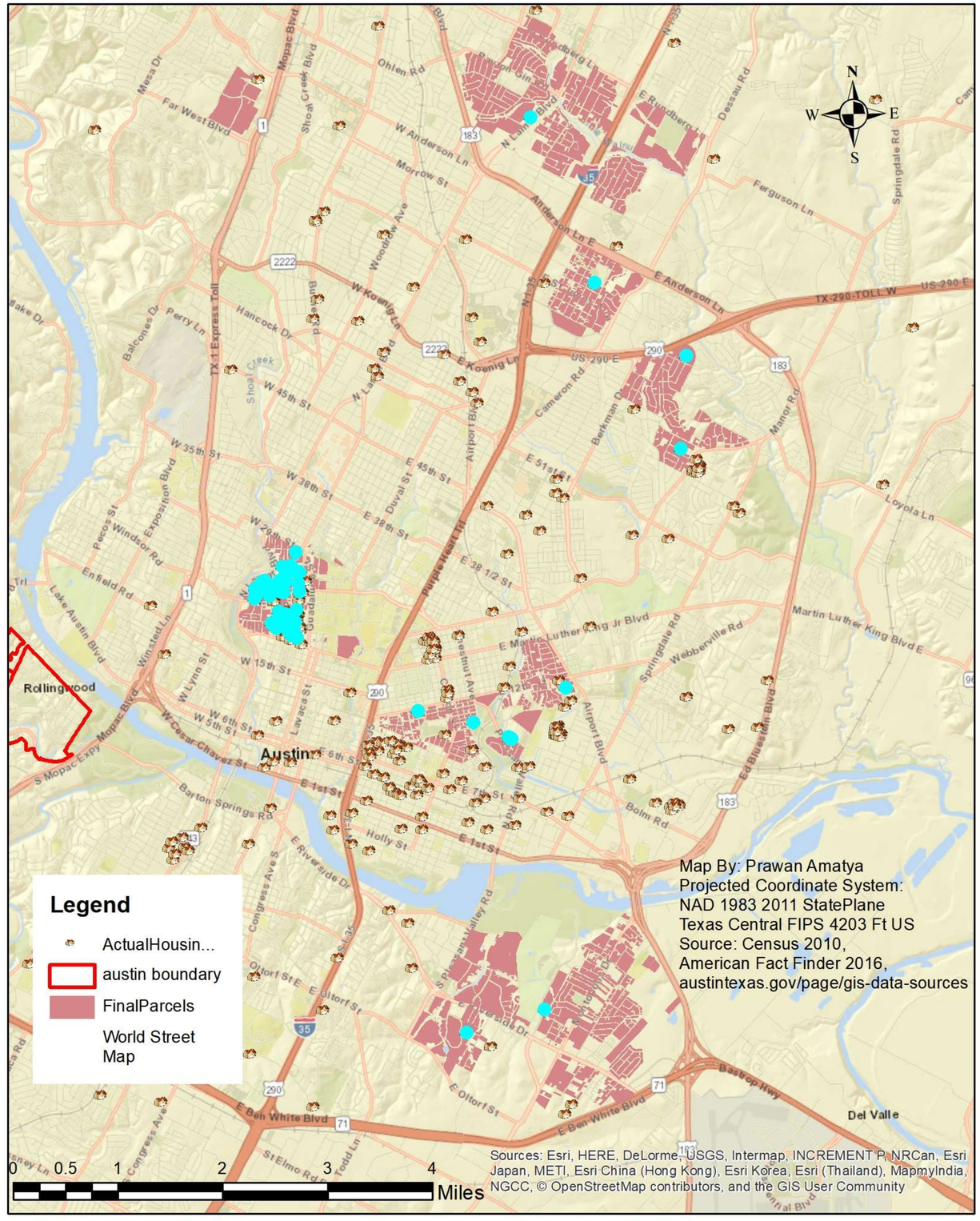
Reference:

https://www.fema.gov/disaster
https://homeguides.sfgate.com/causes-housing-prices-rise-united-states-56413.html
https://www.tdi.texas.gov/reports/documents/harvey-house-data-call-04122018.pdf
_https://www.fema.gov/media-library-data/20130726-1923.../sandy_fema_aar.pdf
https://www.thebalance.com/hurricane-sandy-damage-facts-3305501
Investigation into The Relationship Between Hurricane Storm Parameters and Damage, Jeremy S. Young, The University of North Florida
https://www.hcfcd.org/drainage-network/harris-countys-watersheds/

Affordable Housing on Final Parcels (Austin) with 2 Mile Buffer from Hospital



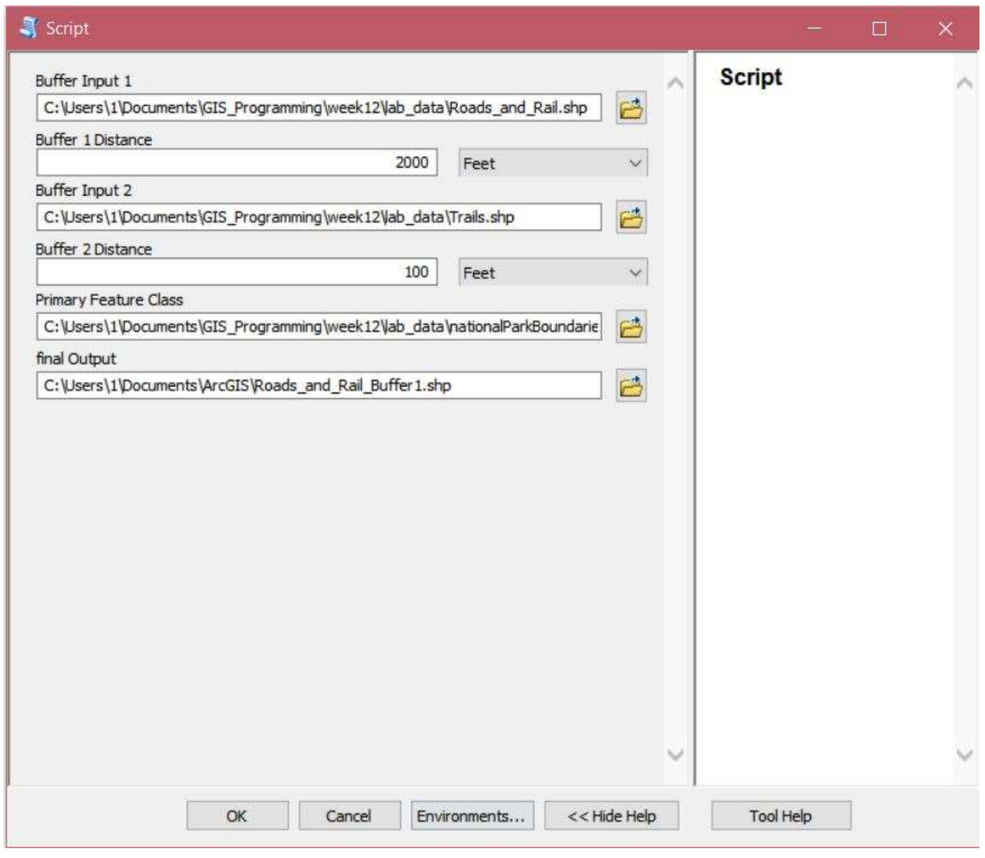
Affordable housing on Final Parcels (Austin)



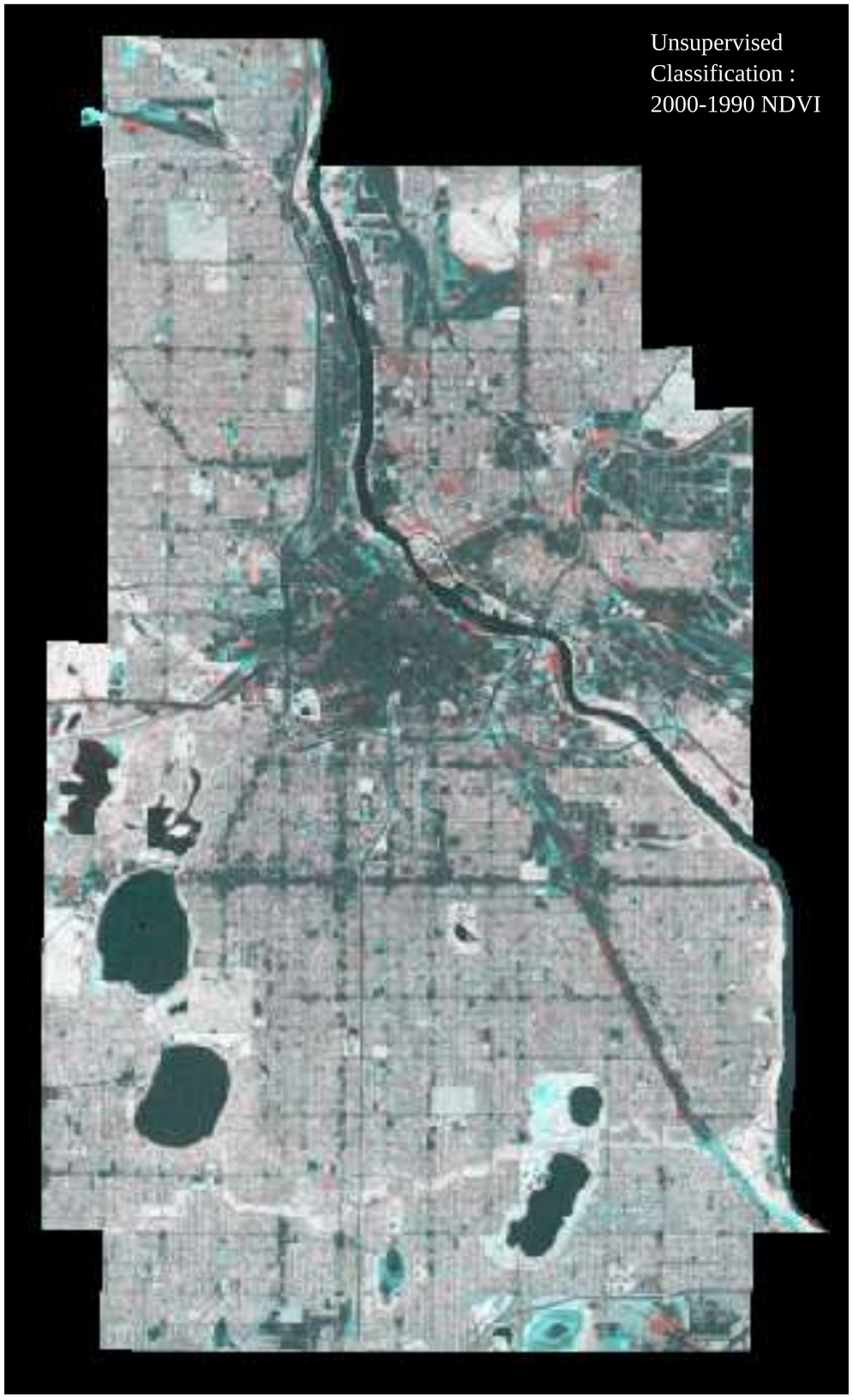
```
# Python Script by Prawan Amatya

‘‘’ Task to create general maps showing roads, streams, and
towns for every county in Texas (there are a lot of counties).
After creating an MXD for the counties, write a Python script
to export that map to a PDF.'‘’

# import required modules
import arcpy

# define the variable for Map Document
mxd = arcpy.mapping.MapDocument( r"C:\Users\1\Documents\SHSU"
r"\GIS_Programming\15\15-1_Script7_data"
r"\Map.mxd" )

# check if the file path is correct
arcpy.AddMessage('mxd opening from ' + mxd.filePath)

# active data frame
adf = mxd.activeDataFrame
print "Active Data Frame is " +adf.name

# dataframe list
df = arcpy.mapping.ListDataFrames(mxd)

# unnest the list
df = df[0]
print df

# Layer object
# listing layers
ll = arcpy.mapping.ListLayers (mxd, '*', df)
# Loop to print the names from the list
for l in ll:
print l.name


# unnest the list
ll=ll[4]
print ll

# title element
te = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT")[0]
print te.text

# use the Search Cursor to sarch rows in the fileds
rows = arcpy.SearchCursor (ll, ["*"])

# loop through and zoom to each polygon
for row in rows:


te.text = row.NAME # gets the row name
print row.name
query = '"FID" = ' +str(row.FID)
#query to get the selection value from the comparision of
# FID of the table with value of the row FID from the
searchCursor
arcpy.SelectLayerByAttribute_management(ll,"NEW_SELECTION",query
)
# select the polygon on the basis of query
df.zoomToSelectedFeatures() # zoomes to selected polygon
df.extent = ll.getSelectedExtent() # Zooms to Layer extent
# define output path and name for map in pdf format
outpdf = r"C:\Users\1\Documents\SHSU\GIS_Programming\15\pdf" \
+ str(row.FID) + str(row.Name) +".pdf"
print outpdf
# to export PDF
arcpy.mapping.ExportToPDF(mxd,outpdf)

del rows #unlock table
del row # unlock row
```

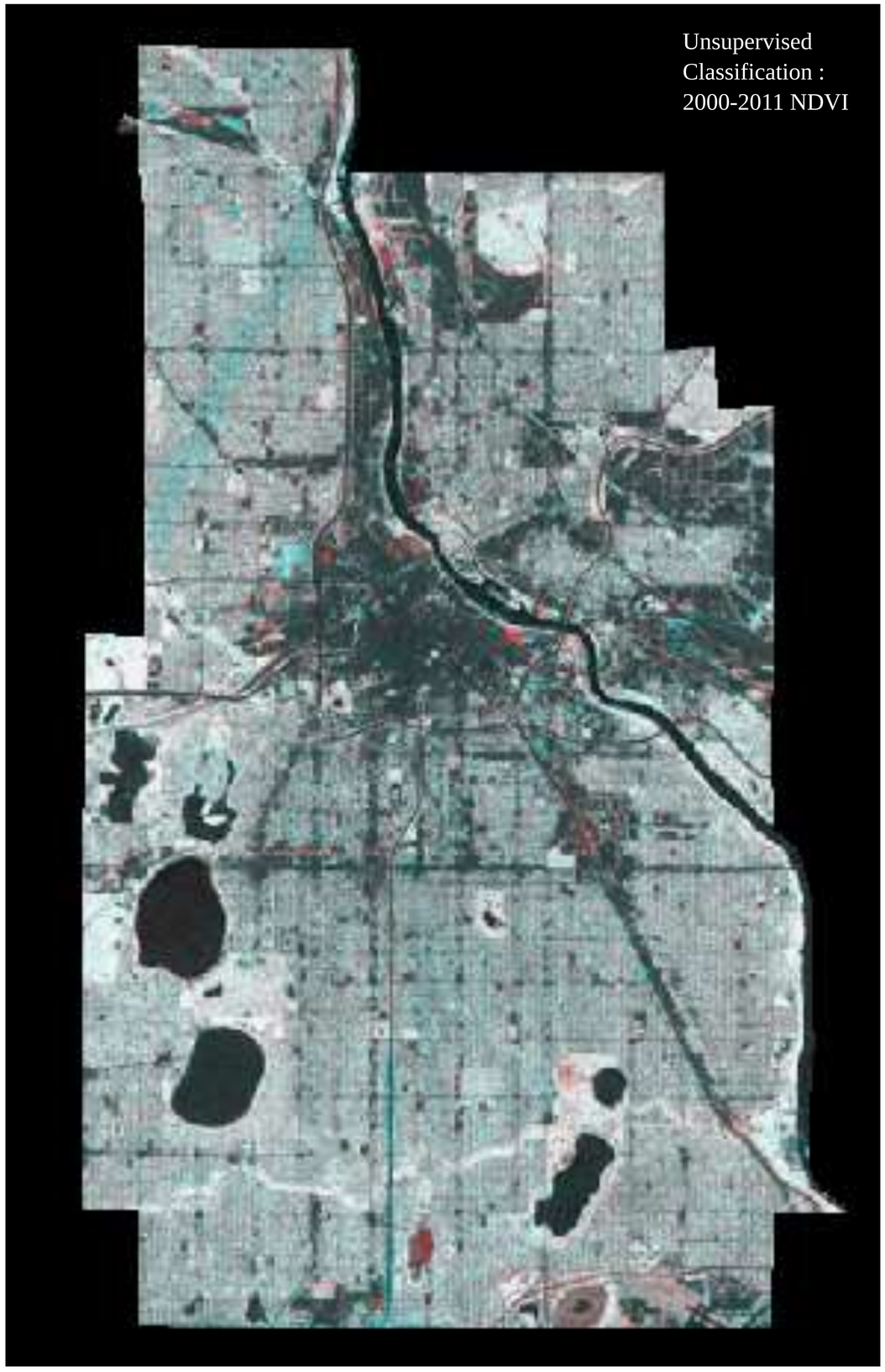
```
# Python Script by : Prawan Amatya

'''The property owned by the company has been surveyed and the data is housed in text
files on the company server. Impress your superiors by writing a Python
script that creates polygons from these files.
The file Week 14 tutorial (attached), contains script and comments
that provide background and examples relevant to this scripting
project. Students will need to copy and paste the text provided in the
tutorial file and save their work as ".py" files.
The data files for this scripting project attached.'''

import arcpy
arcpy.env.workspace = r"C:\Users\1\Documents\SHSU\GIS_Programming\14" \
r"\14_Scrip_data"

arcpy.env.orverwriteOutput = True

# define lists
data =[]
list1 =[]
list2 =[]
list3 =[]
list4 =[]
list5 =[]
list6 =[]


# to open the text file to read the contents
with open ( r"C:\Users\1\Documents\SHSU\GIS_Programming\14"
r"\14_Scrip_data\Propertytest.txt", "r") as f:
data = f.readlines()

name1 = data[1]
name1 = name1[6:]# gets just the name
# gets the name

list1.append(data[2:23])
list1=list1[0] # unnest the lists one level down
list1 = [l.replace('\n','') for l in list1]
# removes the \n character form the list

name2 = data[24]
name2 = name2[6:] # gets just the name
# gets the name

list2.append(data[25:41])
list2 = list2[0] # unnest the lists one level down
list2 = [l.replace('\n', '') for l in list2]
# removes the \n character form the list

name3 = data[43]
name3 = name3[6:] # gets just the name
# gets the name

list3.append(data[44:71])
list3 = list3[0] # unnest the lists one level down
list3 = [l.replace('\n', '') for l in list3]
# removes the \n character form the list

name4 = data[73]
name4 = name4[6:] # gets just the name
# gets the name

list4.append(data[74:88])
list4 = list4[0] # unnest the lists one level down
list4 = [l.replace('\n', '') for l in list4]
# removes the \n character form the list

name5 = data[90]
name5 = name5[6:] # gets just the name
# gets the name

list5.append(data[91:106])
list5 = list5[0] # unnest the lists one level down
list5 = [l.replace('\n', '') for l in list5]
# removes the \n character form the list

name6 = data[108]
name6 = name6[6:] # gets just the name
# gets the name

list6.append(data[109:133])
list6 = list6[0] # unnest the lists one level down
list6 = [l.replace('\n', '') for l in list6]
# removes the \n character form the list


f.close()



# to create feature class shapefile
arcpy.CreateFeatureclass_management( r"C:\Users\1\Documents"
r"\SHSU\GIS_Programming\14"
r"\14_Scrip_data", "polygons.shp"
, "Polygon")

fc ="polygons.shp"

# to add Name field in the attribute table
arcpy.AddField_management("polygons.shp","Name","TEXT","","",16)


### use an insert cursor for this task
# with InsertCursor insures to activate the cursor
with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

# put the list of coordinates in the array as object polygon
for pointXY in list1:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )

cursor.insertRow( [1,polygon, name1] )
# insertRow parameter list follows the index of the InsertCursor
# paramanter list index : ID=1, SHAPE@ = polygon, Name=name1


with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

for pointXY in list2:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )
cursor.insertRow( [2,polygon, name2] )


with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

for pointXY in list3:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )
cursor.insertRow( [3,polygon, name3] )


with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

for pointXY in list4:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )
cursor.insertRow( [4,polygon, name4] )


with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

for pointXY in list5:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )
cursor.insertRow( [5,polygon, name5] )


with arcpy.da.InsertCursor(fc, ["ID","SHAPE@", "Name"]) as cursor:
array = arcpy.Array( )
point = arcpy.Point( )

for pointXY in list6:
point.X, point.Y = pointXY.split( )
array.add( point )
polygon = arcpy.Polygon( array )
cursor.insertRow( [6,polygon, name6] )

# end the cursor
del cursor
```

```
# Prawan Amatya - GIS Programming

'''
Script of GUI tool in ArcMap for the
National Park Service to create a
dataset showing _wilderness_ areas in a
national park. These areas are defined
as 2,000 feet away from a paved road
and 100 feet away from a trail.
To complete this task, buffer zones
needs to be created around the
specified features and then clip those
areas out of a polygon,
showing total coverage of the park.
'''

# project5tool.py

# Import arcpy module
import arcpy
import os
from arcpy import env

# Setup the workspace with overwrite of
existing files
env.overwriteOutput = True

# Get the input argument values form
the tools
input = arcpy.GetParameterAsText(0)

distance = arcpy.GetParameterAsText(1)

input2 = arcpy.GetParameterAsText(2)

distance2= arcpy.GetParameterAsText(3)

input3 = arcpy.GetParameterAsText(4)

finalOutput =
arcpy.GetParameterAsText(5)

# Display progress messages:
arcpy.SetProgressorLabel("Buffer
1.....")

# Process1: Buffer the input feature
class
buff1 = arcpy.Buffer_analysis(input,
"in_memory/buff1", distance, "FULL",
"ROUND",
"ALL", "", "PLANAR")

# Display progress messages:
arcpy.SetProgressorLabel("Buffer
2.....")

buff2 = arcpy.Buffer_analysis(input2,
"in_memory/buff2", distance2, "FULL",
"ROUND", "ALL", "", "PLANAR")


# Display progress messages:
arcpy.SetProgressorLabel("Erase
1.....")

# Process2: Erase
erase1 = arcpy.Erase_analysis(input3,
buff1,
"in_memory/erase1")
# clips out the buffered trails from
National Park


# Display progress messages:
arcpy.SetProgressorLabel("Erase
Final.....")

eraseFinal =
arcpy.Erase_analysis(erase1,
buff2,
finalOutput)

# # nationalParkWidernessArea polygon
is the total coverage of the park
# # that can be marked as wilderness.
arcpy.SetProgressorLabel("Finished")
# print message from tool
```

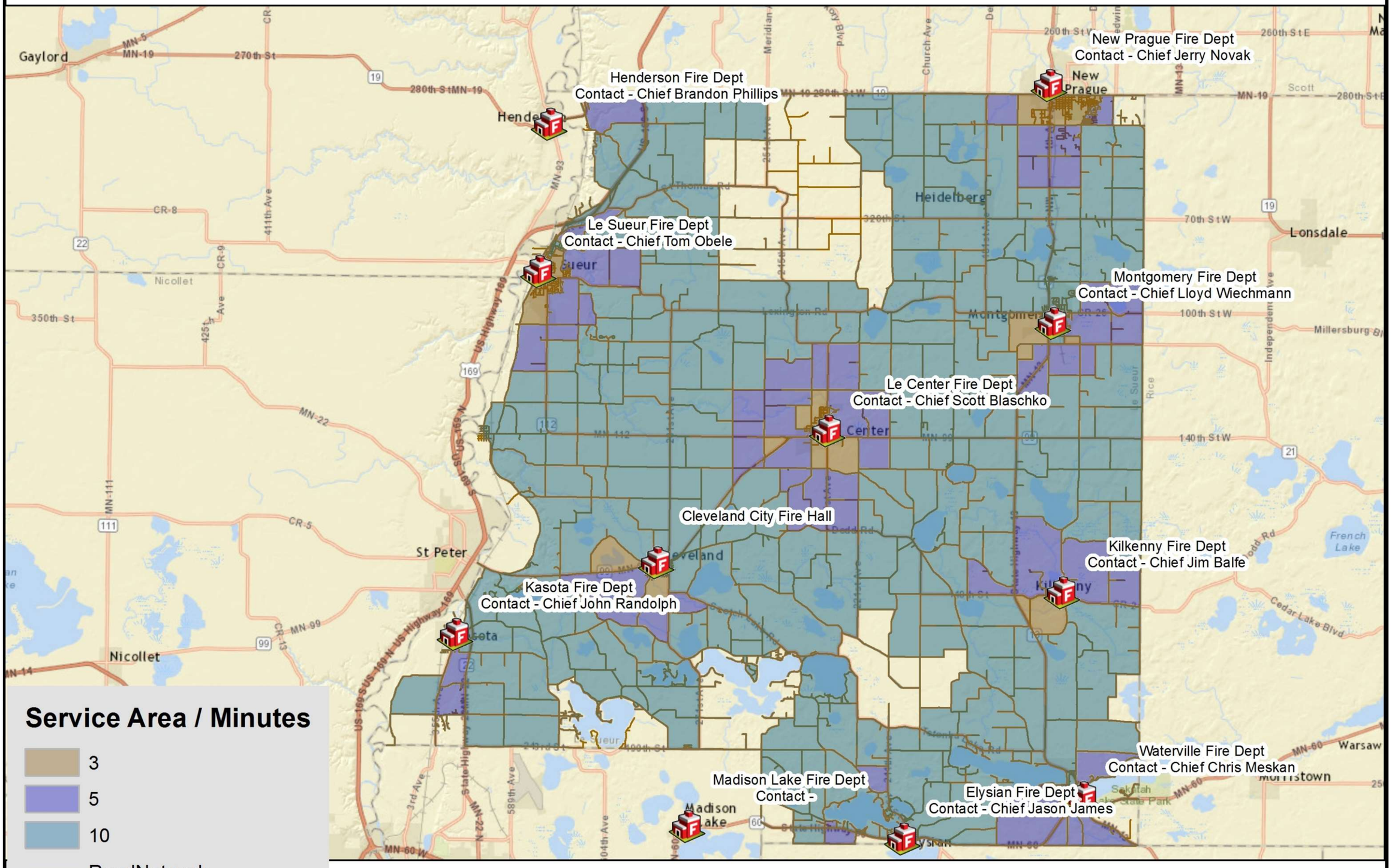


URBAN LAND COVER CHANGE ANALYSIS USING REMOTE SENSING





Le Sueur EMS Service Area

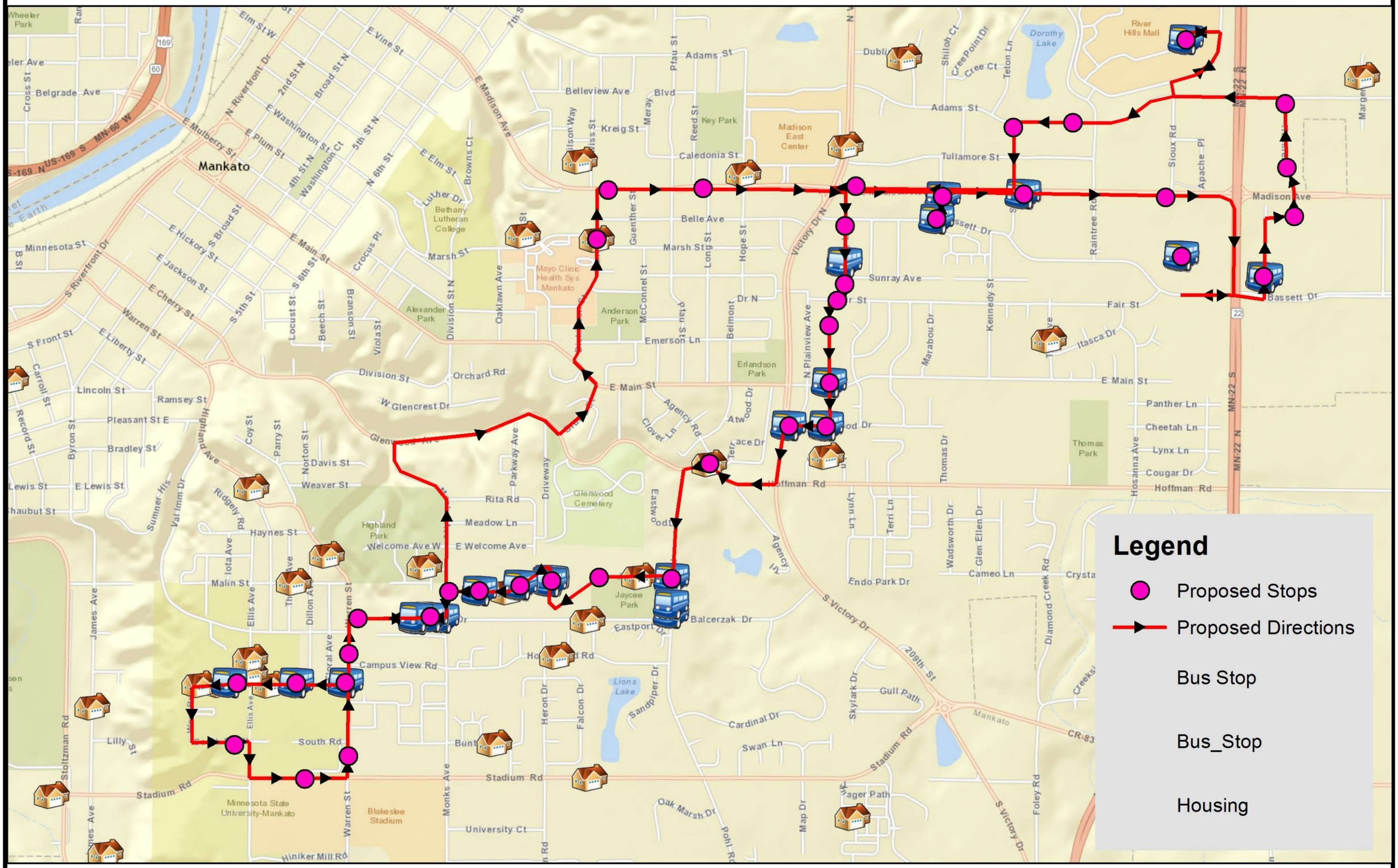


Service Layer Credits: Sources: Esri, HERE, DeLorme, USGS, Intermap, increment P Corp., NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), MapmyIndia, © OpenStreetMap contributors, and the GIS User Community

Prawan Amatya
Intern
Emergency Management
Le Center, Le Sueur County
MN, 56057

N
W E
S

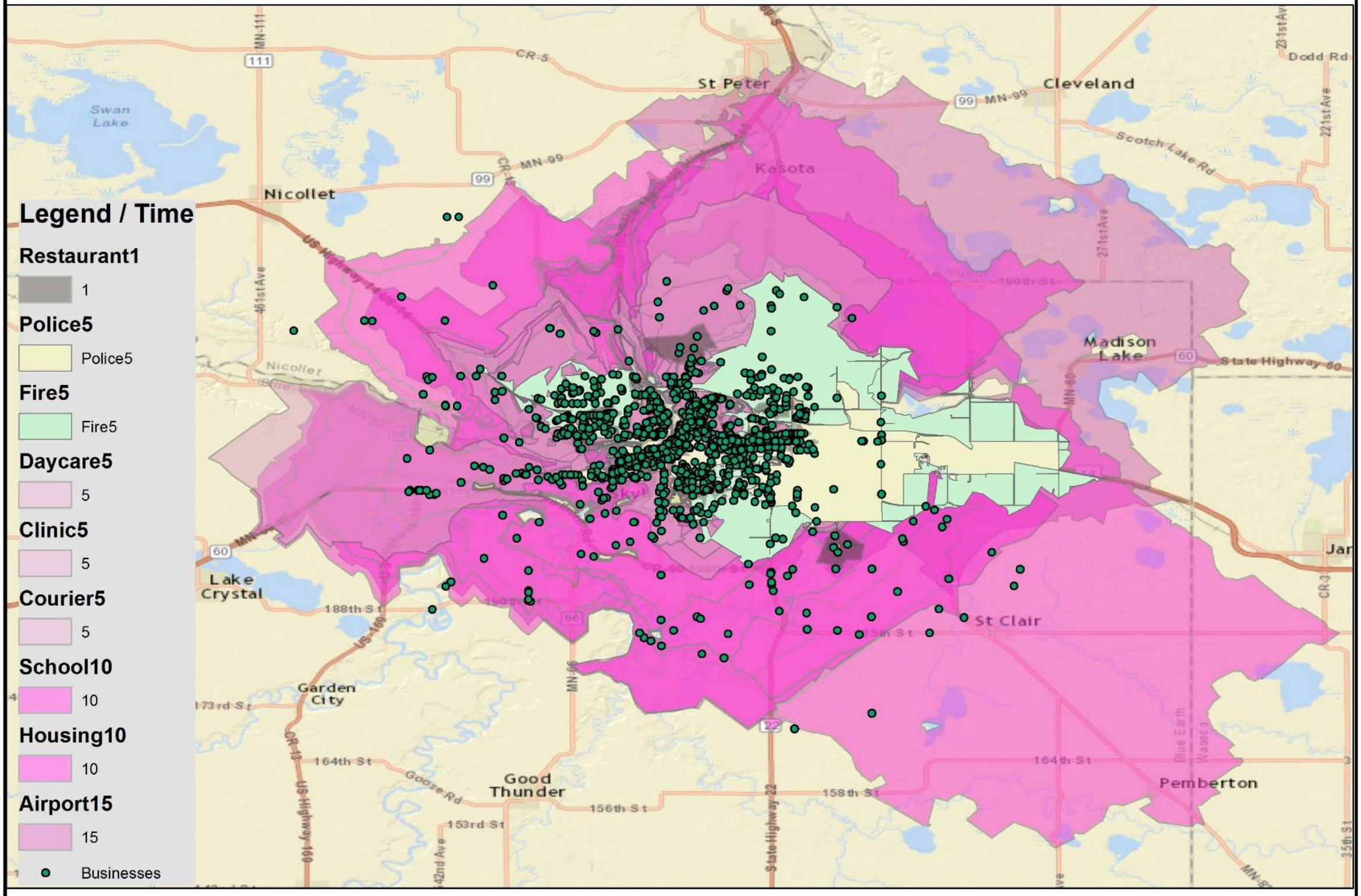
Proposed Route for Minnesota State University Bus



Service Layer Credits: Sources: Esri, HERE, DeLorme, USGS, Intermap, increment P Corp., NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), MapmyIndia, © OpenStreetMap contributors, and the GIS User Community

Prawan Amatya
Graduate GISc
Minnesota State University, Mankato

Best Locations for Office Location Selection Procedure



Service Layer Credits: Sources: Esri, HERE, DeLorme, USGS, Intermap, increment P Corp., NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), MapmyIndia, © OpenStreetMap contributors, and the GIS

0 1.5 3 6 9 12 Miles

Prawan Amatya
Graduate GISc
Minnesota State University
Mankato, 56001

Model Builder: Best Office Location