In [32]:
```
import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [33]:
```
df = pd.read_csv('bangalore-cas-alerts2.csv')
df.head()
```

Out[33]:

	deviceCode_deviceCode	deviceCode_location_latitude	deviceCode_location_longitude	deviceCode_lo
0	8.650000e+14	12.984595	77.744087	
1	8.650000e+14	12.984595	77.744087	
2	8.650000e+14	12.987233	77.741119	
3	8.650000e+14	12.987233	77.741119	
4	8.650000e+14	12.987503	77.740051	

In [34]:
```
df.drop_duplicates(inplace=True)
df.shape
```

Out[34]:
```
(152276, 7)
```

In [35]:
```
# Renaming the columns
columns={
            "deviceCode_deviceCode" : "deviceCode",
            "deviceCode_location_latitude" : "latitude",
            "deviceCode_location_longitude" : "longitude",
            "deviceCode_location_wardName" : "wardName",
            "deviceCode_pyld_alarmType" : "alarmType",
            "deviceCode_pyld_speed" : "speed",
            "deviceCode_time_recordedTime_$date" : "recordedDateTime"
        }

df.rename(columns=columns, inplace=True)
print("Updated column names of train dataframe:", df.columns)
```

```
Updated column names of train dataframe: Index(['deviceCode', 'latitude', 'longitud
e', 'wardName', 'alarmType', 'speed',
       'recordedDateTime'],
      dtype='object')
```

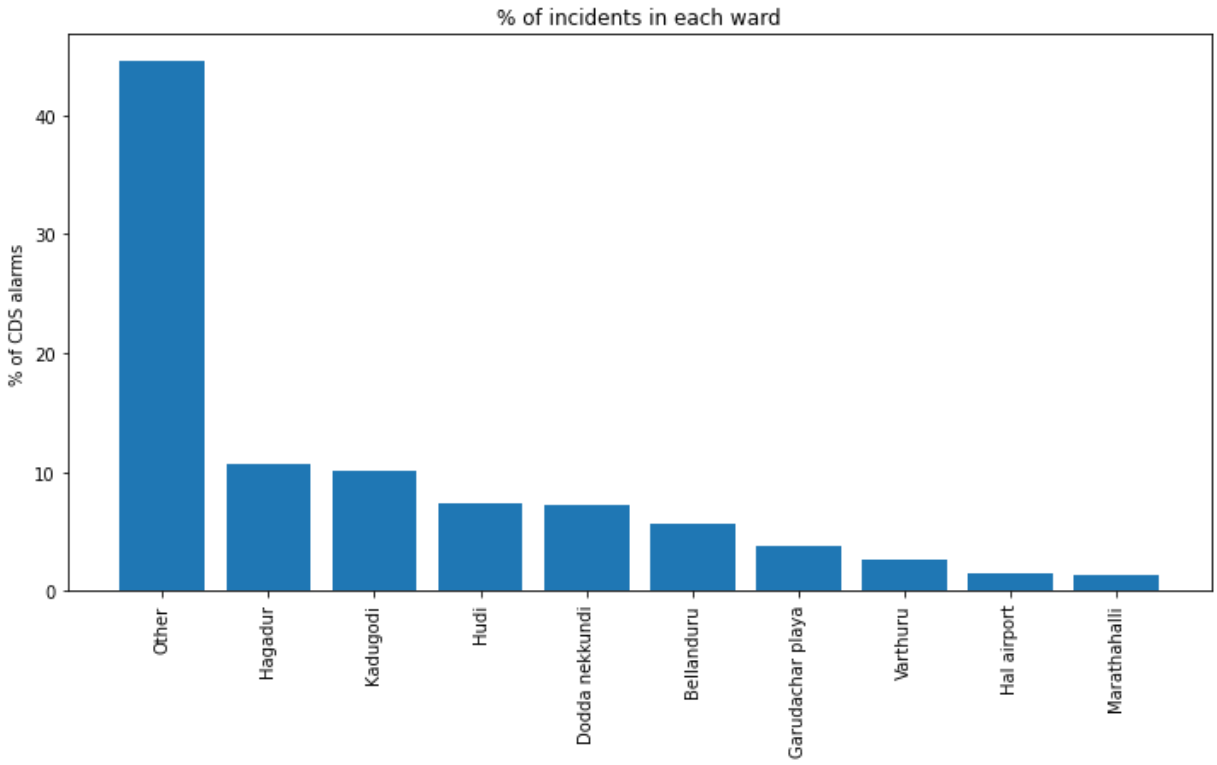
In [36]:
```
lat_max = df.latitude.max()
lat_min = df.latitude.min()
print("Range of latitude:", lat_max, lat_min)
lon_max = df.longitude.max()
lon_min = df.longitude.min()
print("Range of longitude:", lon_max, lon_min)
```

```
Range of latitude: 13.07007504 12.68666267
Range of longitude: 77.80682373 77.50817871
```

In [37]:
```
df['wardName'] = df['wardName'].str.capitalize()
print("Total number of wards in CDS dataset:", len(df['wardName'].unique()))
```

```
Total number of wards in CDS dataset: 49
```

In [38]:
```
fig, axes = plt.subplots(figsize=(12,6))
data = df['wardName'].value_counts(normalize=True).sort_values(ascending=False)
data = data.head(10)
axes.bar(data.index, data*100)
axes.set_ylabel("% of CDS alarms")
axes.set_xticklabels(data.index, rotation=90)
axes.set_title("% of incidents in each ward")
plt.show()
```



In [39]:
```
bbmp_data = pd.read_csv('BBMP.csv')
bbmp_data.head()
```

Out[39]:

	X	Y	KGISWardID	KGISWardCode	LGD_WardCode	KGISWardNo	KGISWardName
0	77.601487	13.110062	4878	2003001	1303139.0	1	Kempegowda Ward
1	77.575024	13.122529	4879	2003002	1303140.0	2	Chowdeswari Ward
2	77.577280	13.101535	4882	2003003	1303141.0	3	Someshwara Ward
3	77.555032	13.097807	4883	2003004	1303142.0	4	Atturu Layout
4	77.588609	13.090581	4886	2003005	1303143.0	5	Yelahanka Satellite Town

In [40]:

```
# Capitalize ward name
bbmp_data['KGISWardName'] = bbmp_data['KGISWardName'].str.capitalize()
print("Total number of wards in BBMP dataset:", len(bbmp_data['KGISWardName'].unique
# Create a dict mapping of ward number and names
ward_numbers = bbmp_data['KGISWardNo'].unique()
ward_names = bbmp_data['KGISWardName'].unique()
ward_dict = dict(zip(ward_numbers, ward_names))
```

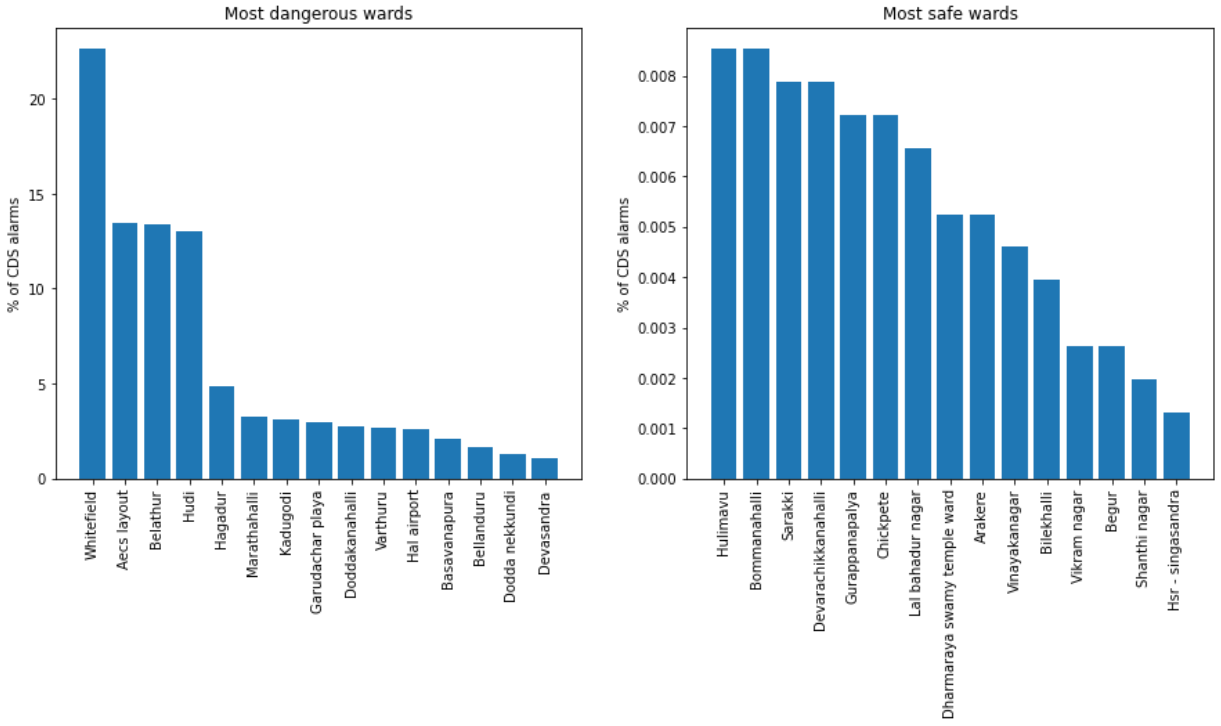
```
Total number of wards in BBMP dataset: 243
```

In [41]:

```
from sklearn.neighbors import KNeighborsClassifier

knn_clf = KNeighborsClassifier(n_neighbors=1, weights='distance')
knn_clf.fit(bbmp_data[['Y', 'X']], bbmp_data['KGISWardNo'])

# Estimate ward no in train_data from learnt model of bbmp_data
df['estimatedWardNo'] = knn_clf.predict(df[['latitude', 'longitude']])
# Estimate ward name in train_data from the ward no - ward name mapping dictionary
df['estimatedWardName'] = df['estimatedWardNo'].map(ward_dict)
```

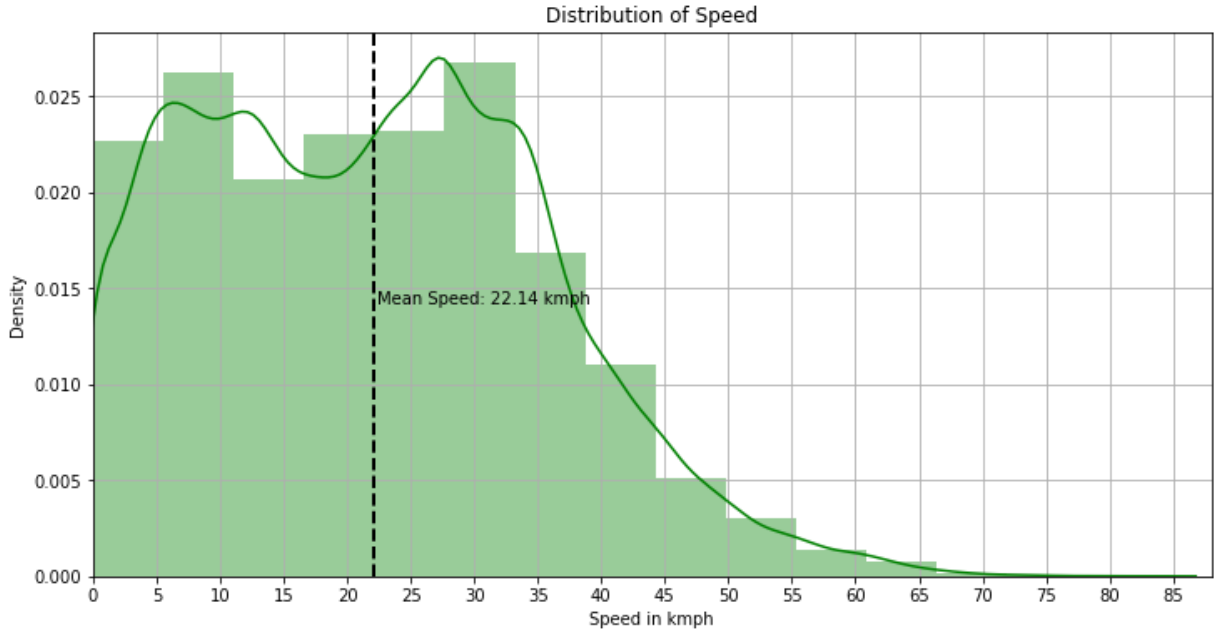
In [42]:

```
fig, [axes1, axes2] = plt.subplots(1, 2, figsize=(15,6))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values(ascending=Fa
data1 = data.head(15)

axes1.bar(data1.index, data1*100)
axes1.set_ylabel("% of CDS alarms")
axes1.set_xticklabels(data1.index, rotation=90)
axes1.set_title("Most dangerous wards")

data2 = data.tail(15)
axes2.bar(data2.index, data2*100)
axes2.set_ylabel("% of CDS alarms")
axes2.set_xticklabels(data2.index, rotation=90)
axes2.set_title("Most safe wards")

plt.show()
```



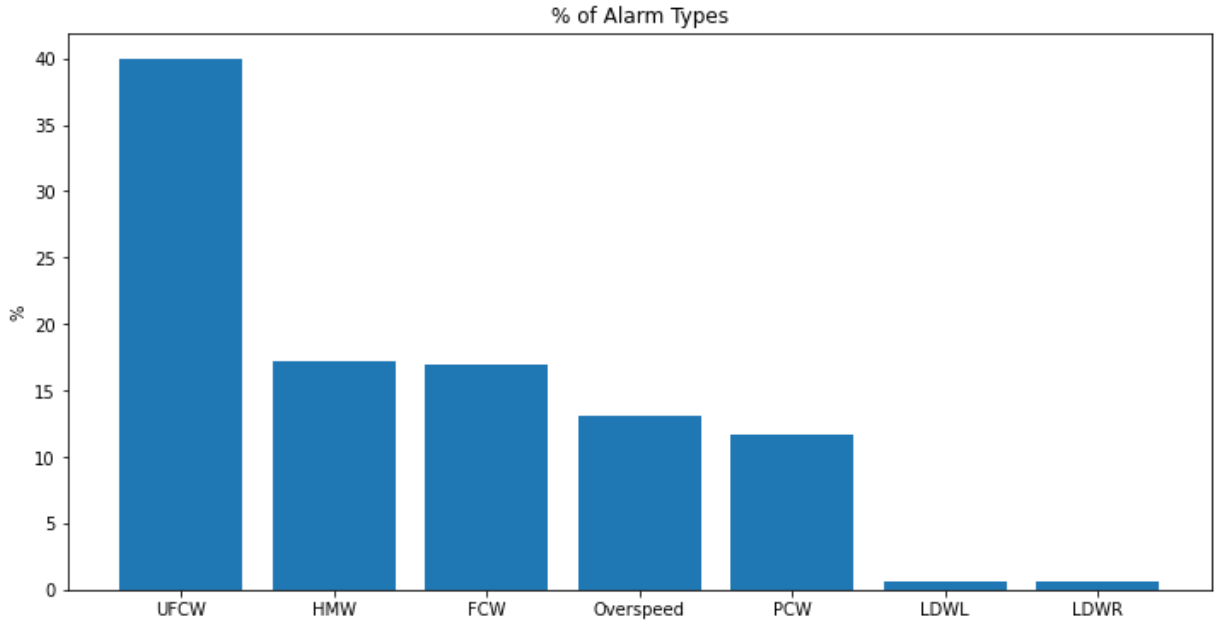
In [43]:

```
fig, axes = plt.subplots(figsize=(12,6))
data = df['speed']
#axes.hist(data, bins=15, color='green')
sns.distplot(data, bins=15, color='green')
axes.axvline(data.mean(), color='k', linestyle='dashed', linewidth=2)

axes.set_xticks(np.arange(0, data.max()+5, 5))
axes.set_xticklabels([str(val) for val in np.arange(0, data.max()+5, 5)])
axes.set_xlim(0, data.max()+5)
axes.set_xlabel('Speed in kmph')
axes.set_title('Distribution of Speed')
axes.grid(True)

_ymin, _ymax = axes.get_ylim()
axes.text(data.mean() + data.mean()/100,
          (_ymax+_ymin)*0.5,
          'Mean Speed: {:.2f} kmph'.format(data.mean()))

plt.show()
```



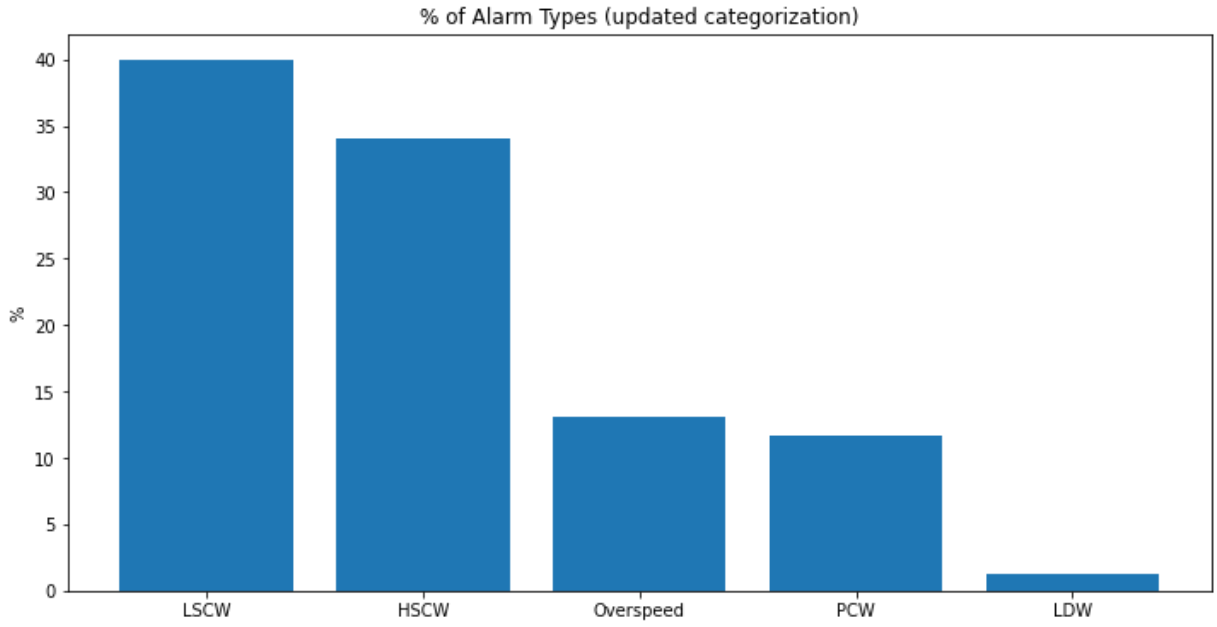
In [44]:

```
fig, axes = plt.subplots(figsize=(12,6))
data = df['alarmType'].value_counts(normalize=True)
axes.bar(data.index, data*100)
axes.set_title('% of Alarm Types')
axes.set_xlabel('')
axes.set_ylabel('%')
plt.show()
```



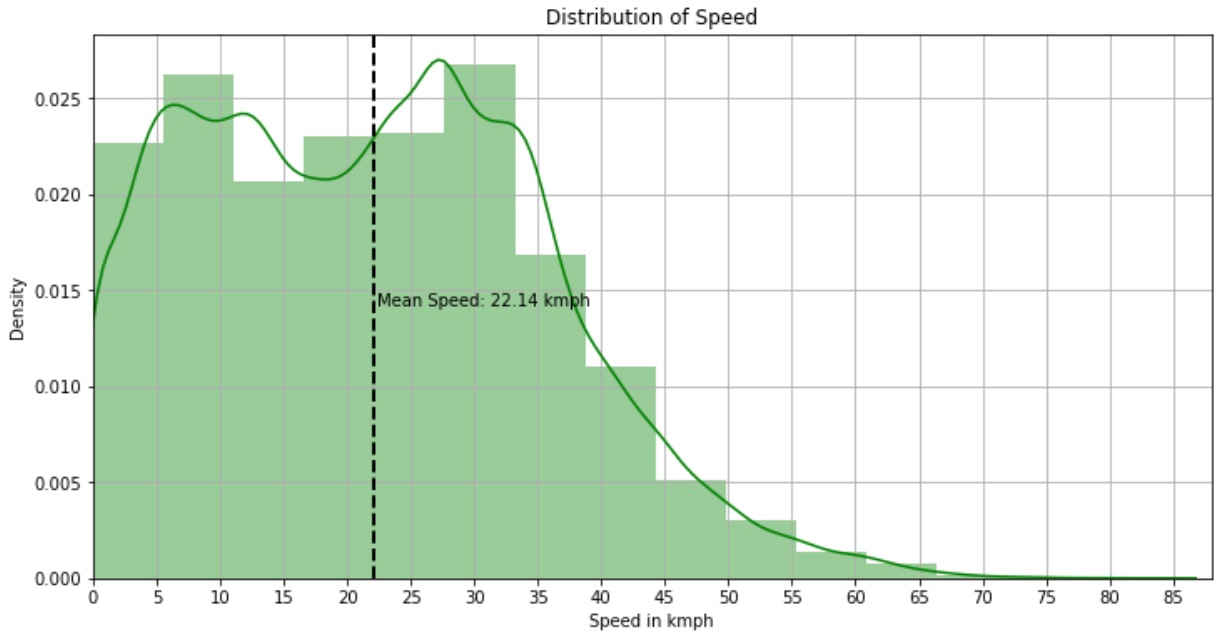
In [45]:
```
alarm_mapping = {"UFCW":"LSCW", "HMW":"HSCW", "FCW":"HSCW", "LDWL":"LDW", "LDWR":"LD
df['alarmTypeCat'] = df['alarmType'].map(alarm_mapping)
```

In [46]:
```
fig, axes = plt.subplots(figsize=(12,6))
data = df['alarmTypeCat'].value_counts(normalize=True)
axes.bar(data.index, data*100)
axes.set_title('% of Alarm Types (updated categorization)')
axes.set_xlabel('')
axes.set_ylabel('%')
plt.show()
```



In [47]:
```
fig, axes = plt.subplots(figsize=(12,6))
data = df['speed']
#axes.hist(data, bins=15, color='green')
sns.distplot(data, bins=15, color='green')
axes.axvline(data.mean(), color='k', linestyle='dashed', linewidth=2)

axes.set_xticks(np.arange(0, data.max()+5, 5))
axes.set_xticklabels([str(val) for val in np.arange(0, data.max()+5, 5)])
axes.set_xlim(0, data.max()+5)
```

```
axes.set_xlabel('Speed in kmph')
axes.set_title('Distribution of Speed')
axes.grid(True)

_ymin, _ymax = axes.get_ylim()
axes.text(data.mean() + data.mean()/100,
          (_ymax+_ymin)*0.5,
          'Mean Speed: {:.2f} kmph'.format(data.mean()))

plt.show()
```

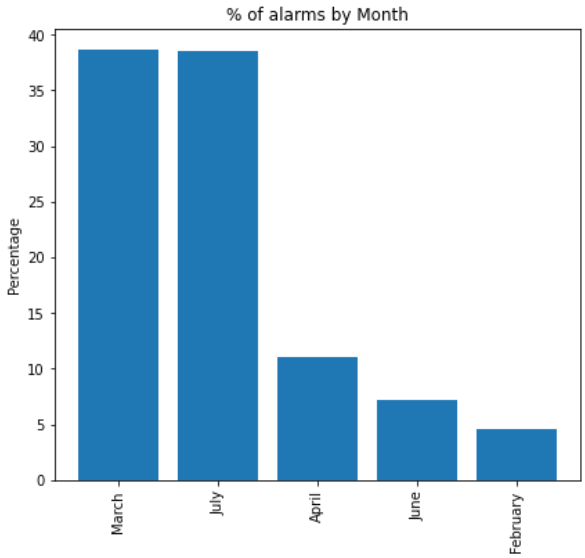


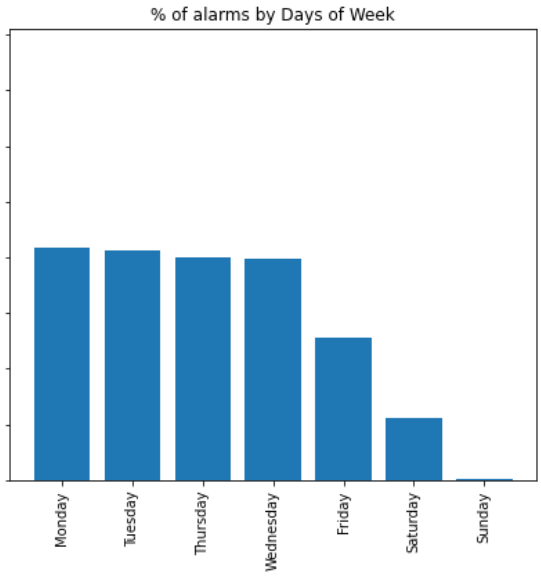
In [48]:
```
df.recordedDateTime = df.recordedDateTime.map(lambda x : pd.Timestamp(x, tz='Asia/Ko
```

In [49]:
```
print("Alarm data are spanning across:")
print("Years: ", df.recordedDateTime.dt.year.unique())
print("Months: ", df.recordedDateTime.dt.month_name().unique())
print("Dates: ", df.recordedDateTime.dt.day.unique())
```

```
Alarm data are spanning across:
Years:  [2018]
Months:  ['February' 'March' 'April' 'June' 'July']
Dates:  [ 1  2  3  5  6  7  8  9 12 14 15 16 19 20 21 22 23 24 25 26 27 28 10 13
 17 29 31  4 11 18 30]
```

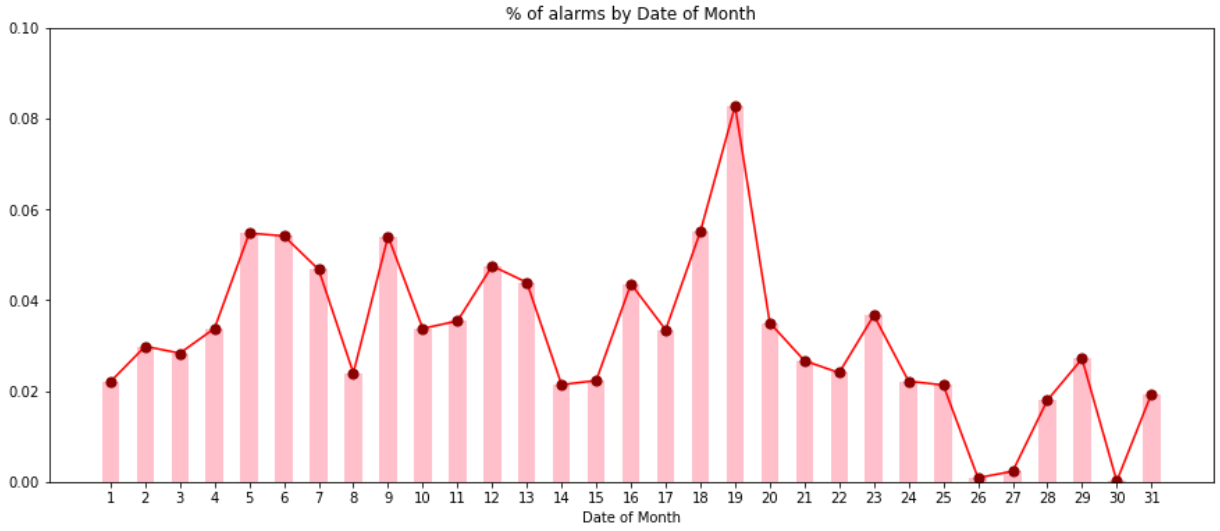
In [50]:
```
fig, [axes1, axes2] = plt.subplots(1, 2, figsize=(15,6), sharey=True)
df["monthName"] = df.recordedDateTime.dt.month_name()
data = df["monthName"].value_counts(normalize=True)

axes1.bar(data.index, data*100)
axes1.set_xticklabels(data.index, rotation=90)
axes1.set_ylabel('Percentage')
axes1.set_title('% of alarms by Month')

df["dayName"] = df.recordedDateTime.dt.day_name()
data = df["dayName"].value_counts(normalize=True)
axes2.bar(data.index, data*100)
axes2.set_xticklabels(data.index, rotation=90)
axes2.set_title('% of alarms by Days of Week')

plt.show()
```





In [51]:

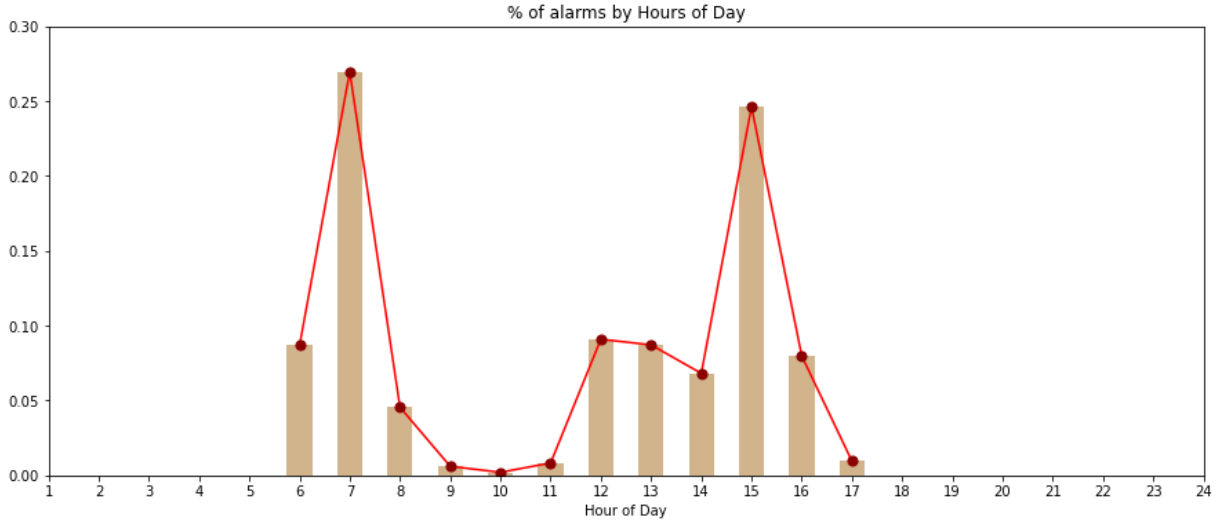
```
fig, axes = plt.subplots(figsize=(15,6))
df["dayOfMonth"] = df.recordedDateTime.dt.day
data = df["dayOfMonth"].value_counts(normalize=True).sort_index()
axes.bar(data.index, data, color='pink', width=0.5, zorder=0)
axes.plot(data.index, data, color='red', zorder=1)
axes.scatter(data.index, data, s=50, color='darkred', zorder=2)

axes.set_xlabel('Date of Month')
axes.set_xticks(np.arange(1, 32))
axes.set_xticklabels([str(val) for val in np.arange(1, 32)])
axes.set_ylim(0, 0.1)
axes.set_title('% of alarms by Date of Month')

plt.show()
```



In [52]:

```
fig, axes = plt.subplots(figsize=(15,6))
df["hour"] = df.recordedDateTime.dt.hour
data = df["hour"].value_counts(normalize=True).sort_index()

axes.bar(data.index, data, color='tan', width=0.5, zorder=0)
axes.plot(data.index, data, color='red', zorder=1)
axes.scatter(data.index, data, s=50, color='darkred', zorder=2)

axes.set_xlabel('Hour of Day')
axes.set_xticks(np.arange(1, 25))
axes.set_xticklabels([str(val) for val in np.arange(1, 25)])
```

```
axes.set_ylim(0, 0.3)
axes.set_title('% of alarms by Hours of Day')

plt.show()
```



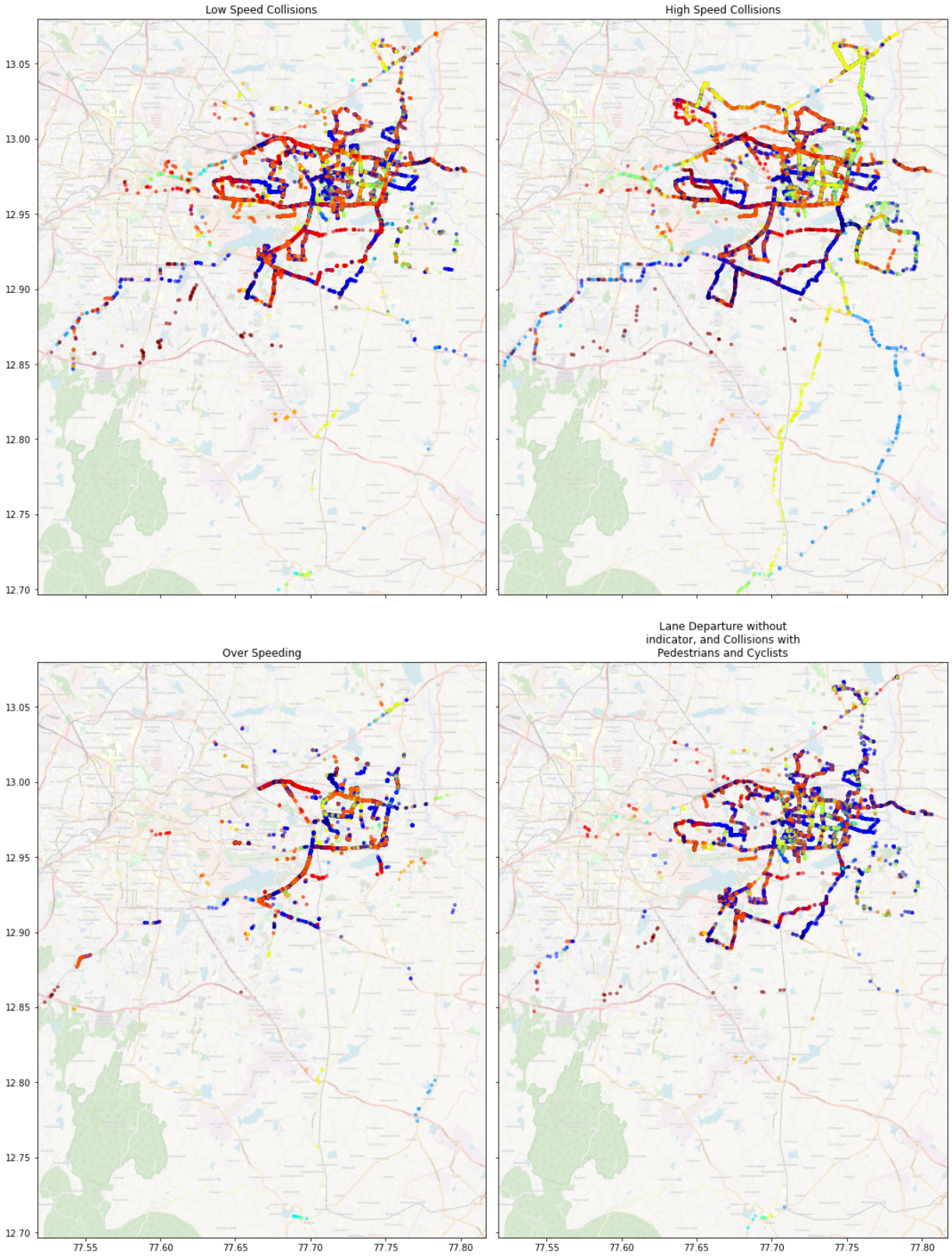
In [53]:

```
data_lscw = df[df.alarmTypeCat == 'LSCW']
data_hscw = df[df.alarmTypeCat == 'HSCW']
data_speed = df[df.alarmTypeCat == 'Overspeed']
data_pcw = df[df.alarmTypeCat.str.contains('PCW', 'LDW')]
```

In [54]:

```
from textwrap import wrap
bangalore_map_img = 'https://lh3.googleusercontent.com/np8igtYRrHpe7rvJwMzVhbyUZC4Np
bangalore_map = plt.imread(bangalore_map_img)
fig, axes = plt.subplots(2, 2, figsize=(15,20), sharex=True, sharey=True)
cmap = plt.get_cmap("jet")

# Plot LSCW
axes[0][0].scatter(data_lscw.longitude, data_lscw.latitude,
                   alpha=0.5, marker="o", s=10,
                   c=data_lscw.hour, cmap=cmap, zorder=1)
axes[0][0].set_title("Low Speed Collisions")

# Plot HSCW
axes[0][1].scatter(data_hscw.longitude, data_hscw.latitude,
                   alpha=0.5, marker="o", s=10,
                   c=data_hscw.hour, cmap=cmap, zorder=1)
axes[0][1].set_title("High Speed Collisions")

# Plot Overspeed
axes[1][0].scatter(data_speed.longitude, data_speed.latitude,
                   alpha=0.5, marker="o", s=10,
                   c=data_speed.hour, cmap=cmap, zorder=1)
axes[1][0].set_title("Over Speeding")

# Plot PCW and LDW
axes[1][1].scatter(data_pcw.longitude, data_pcw.latitude,
                   alpha=0.5, marker="o", s=10,
                   c=data_pcw.hour, cmap=cmap, zorder=1)
axes[1][1].set_title("\n".join(wrap("Lane Departure without indicator, and Collision

# Plot Bangalore map image
epsilon = 0.01
bound_box = [lon_min + epsilon, lon_max + epsilon,
             lat_min + epsilon, lat_max + epsilon]
```

```
for ax in axes.flat:
    im = ax.imshow(bangalore_map, extent=bound_box,
                   alpha=0.5, zorder=0, cmap=cmap)

plt.tight_layout()
plt.show()
```

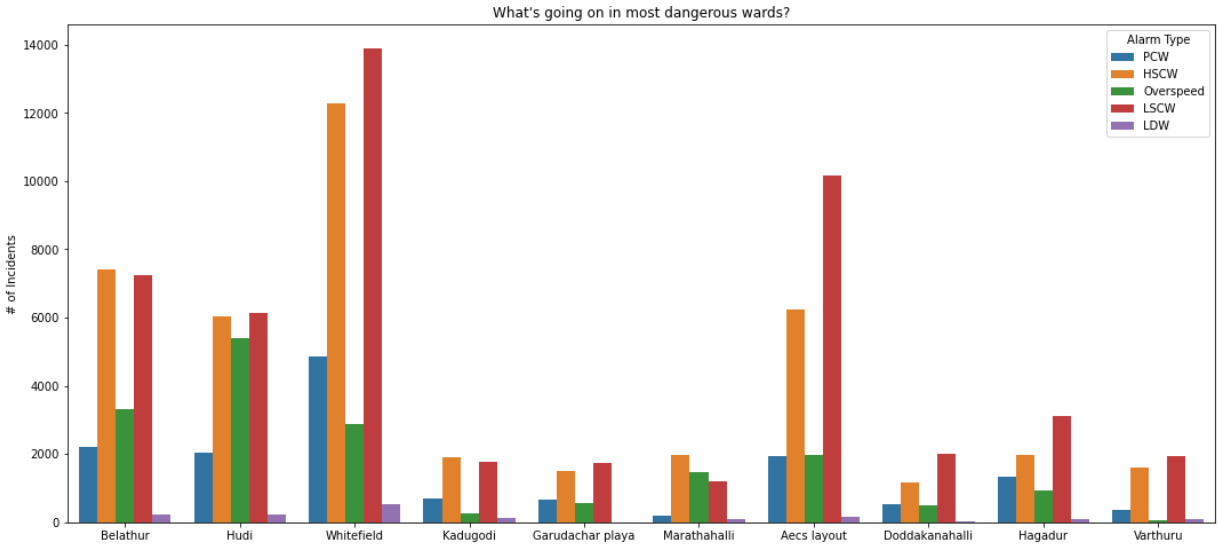


In [61]:

```
fig, axes = plt.subplots(figsize=(18,8))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values(ascending=Fa
data = data.head(10)
print(data)
ward_top = data.index
```

```
ward_top_data =df[df.estimatedWardName.isin(ward_top)]
sns.countplot(x='estimatedWardName', hue='alarmTypeCat', data=ward_top_data, ax=axes

axes.legend(title='Alarm Type')
axes.set_xlabel('')
axes.set_ylabel('# of Incidents')
axes.set_title('What\'s going on in most dangerous wards?')

plt.show()
```

```
Whitefield          0.226385
Aecs layout         0.134578
Belathur            0.134145
Hudi                0.130369
Hagadur             0.048852
Marathahalli        0.032507
Kadugodi            0.031266
Garudachar playa    0.029525
Doddakanahalli      0.027614
Varthuru            0.026642
Name: estimatedWardName, dtype: float64
```

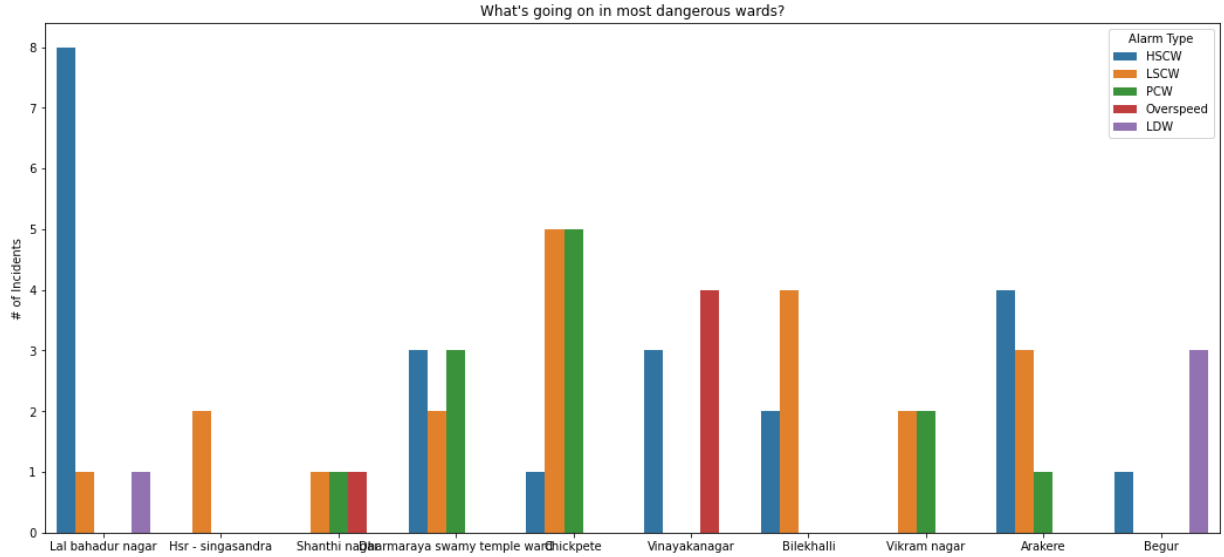


In [62]:

```
fig, axes = plt.subplots(figsize=(18,8))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values()
data = data.head(10)
print(data)
ward_top = data.index

ward_top_data =df[df.estimatedWardName.isin(ward_top)]
sns.countplot(x='estimatedWardName', hue='alarmTypeCat', data=ward_top_data, ax=axes

axes.legend(title='Alarm Type')
axes.set_xlabel('')
axes.set_ylabel('# of Incidents')
axes.set_title('What\'s going on in most dangerous wards?')

plt.show()
```

```
Hsr - singasandra              0.000013
Shanthi nagar                  0.000020
Vikram nagar                   0.000026
Begur                          0.000026
Bilekhalli                     0.000039
Vinayakanagar                  0.000046
Arakere                        0.000053
```

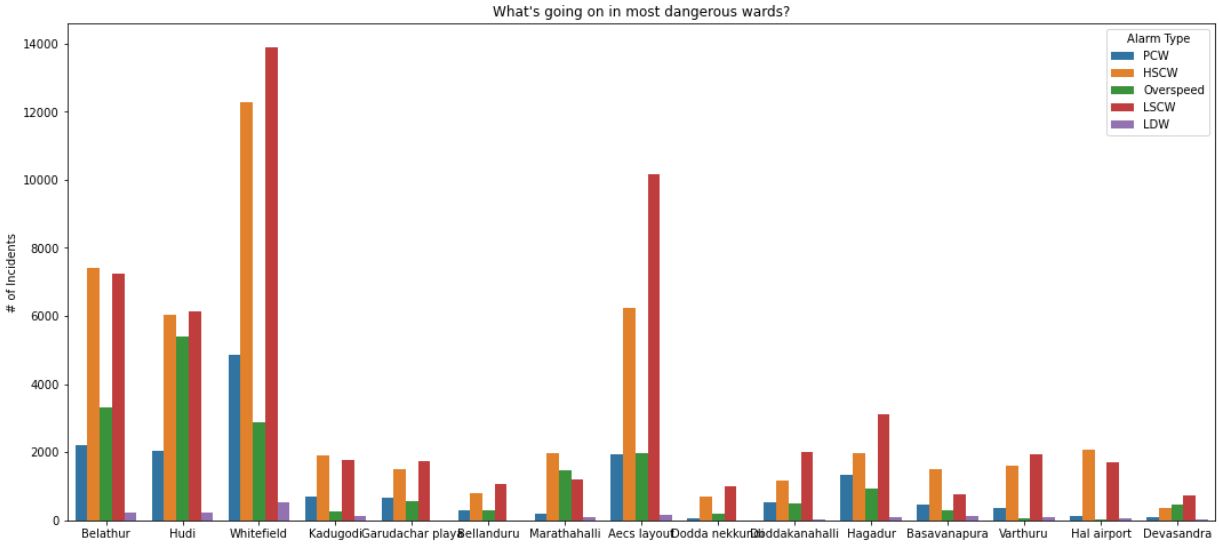
```
Dharmaraya swamy temple ward    0.000053
Lal bahadur nagar               0.000066
Chickpete                       0.000072
Name: estimatedWardName, dtype: float64
```



In [63]:

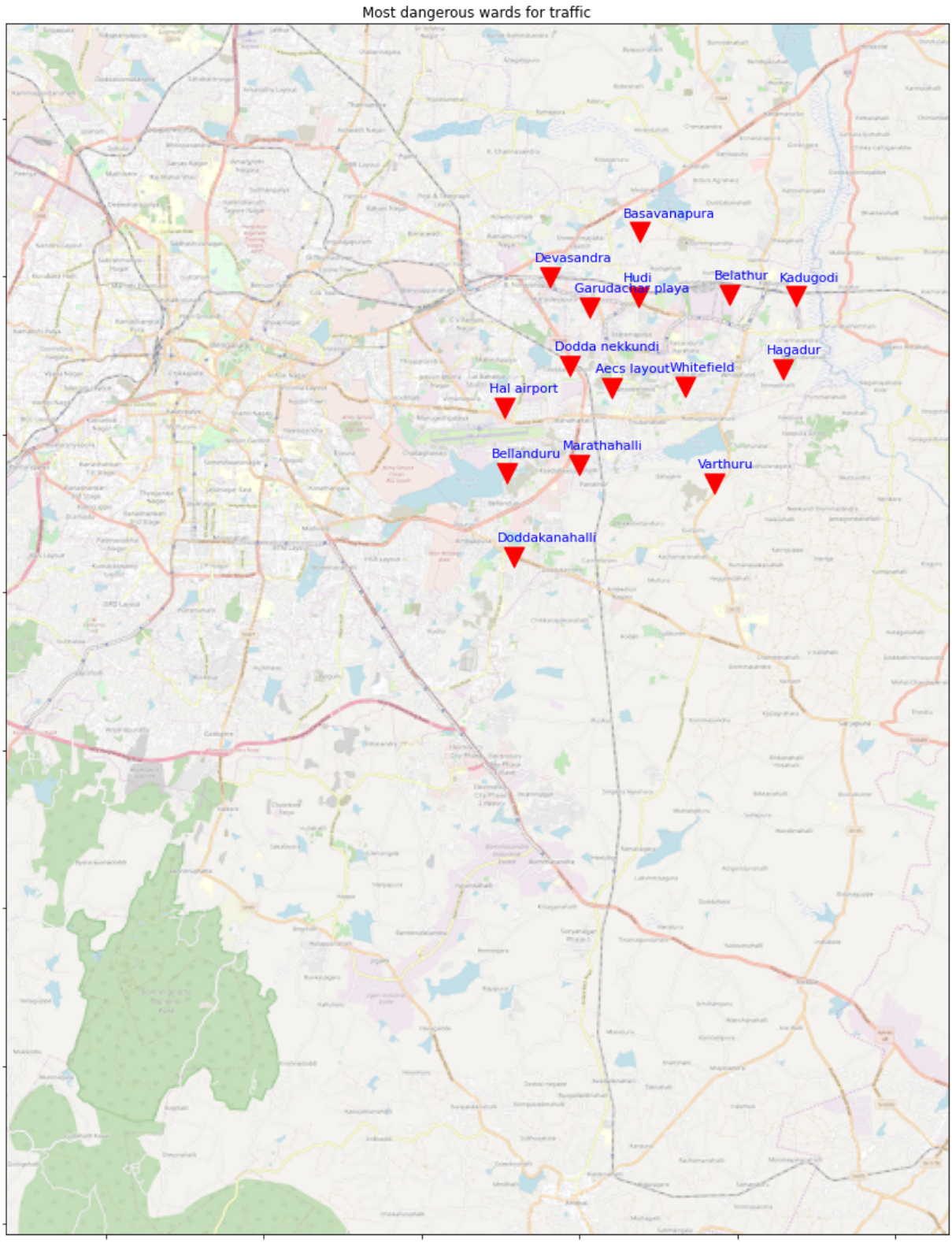
```
fig, axes = plt.subplots(figsize=(18,8))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values(ascending=Fa
data = data.head(15)
print(data)
ward_top = data.index

ward_top_data =df[df.estimatedWardName.isin(ward_top)]
sns.countplot(x='estimatedWardName', hue='alarmTypeCat', data=ward_top_data, ax=axes

axes.legend(title='Alarm Type')
axes.set_xlabel('')
axes.set_ylabel('# of Incidents')
axes.set_title('What\'s going on in most dangerous wards?')

plt.show()
```

```
Whitefield          0.226385
Aecs layout         0.134578
Belathur            0.134145
Hudi                0.130369
Hagadur             0.048852
Marathahalli        0.032507
Kadugodi            0.031266
Garudachar playa    0.029525
Doddakanahalli      0.027614
Varthuru            0.026642
Hal airport         0.026307
Basavanapura        0.020771
Bellanduru          0.016398
Dodda nekkundi      0.012858
Devasandra          0.010941
Name: estimatedWardName, dtype: float64
```



In [64]:

```
fig, axes = plt.subplots(figsize=(15,15))
cmap = plt.get_cmap("jet")

data = bbmp_data[bbmp_data.KGISWardName.isin(ward_top)]

axes.scatter(data.X, data.Y, marker="v", s=300,
                  c='red', zorder=1)
axes.set_title("Most dangerous wards for traffic")

# Plot Bangalore map image
epsilon = 0.01
bound_box = [lon_min + epsilon, lon_max + epsilon,
             lat_min + epsilon, lat_max + epsilon]

axes.imshow(bangalore_map, extent=bound_box, alpha=0.7, zorder=0)

# Add names of wards as text
for _idx, _ward_data in data.iterrows():
    axes.text(_ward_data.X  - epsilon/2, _ward_data.Y + epsilon/2,
              _ward_data.KGISWardName, color='blue', fontsize=11)

axes.set_yticklabels([])
axes.set_xticklabels([])

plt.tight_layout()
plt.show()
```



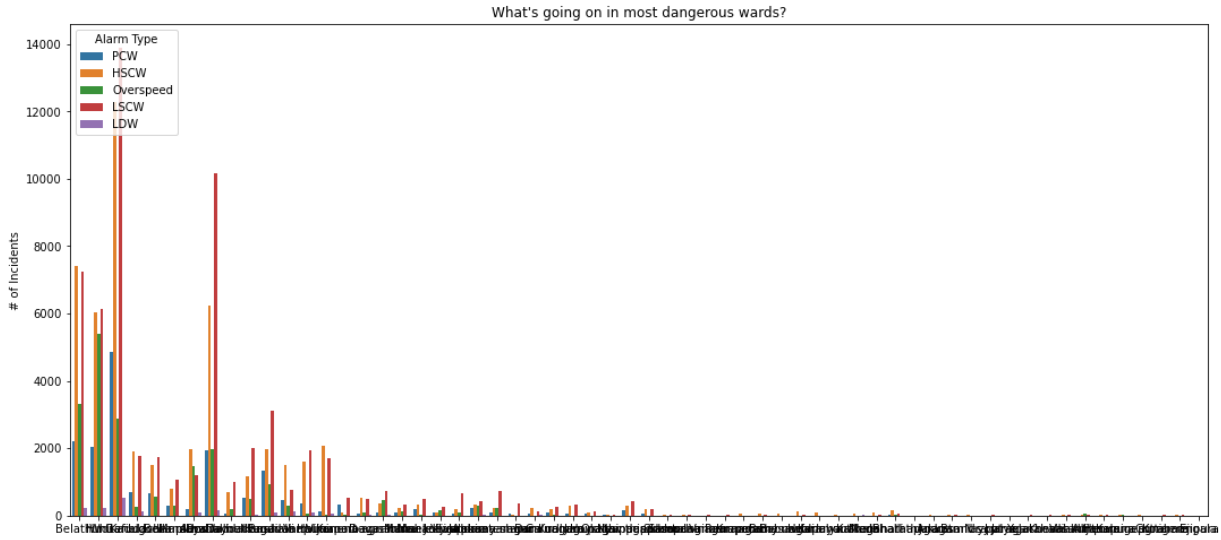
In [65]:

```
fig, axes = plt.subplots(figsize=(18,8))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values(ascending=Fa
data = data.head(60)

print(data)
ward_top = data.index

ward_top_data =df[df.estimatedWardName.isin(ward_top)]
sns.countplot(x='estimatedWardName', hue='alarmTypeCat', data=ward_top_data, ax=axes

axes.legend(title='Alarm Type')
axes.set_xlabel('')
axes.set_ylabel('# of Incidents')
```

```
axes.set_title('What\'s going on in most dangerous wards?')

plt.show()
```

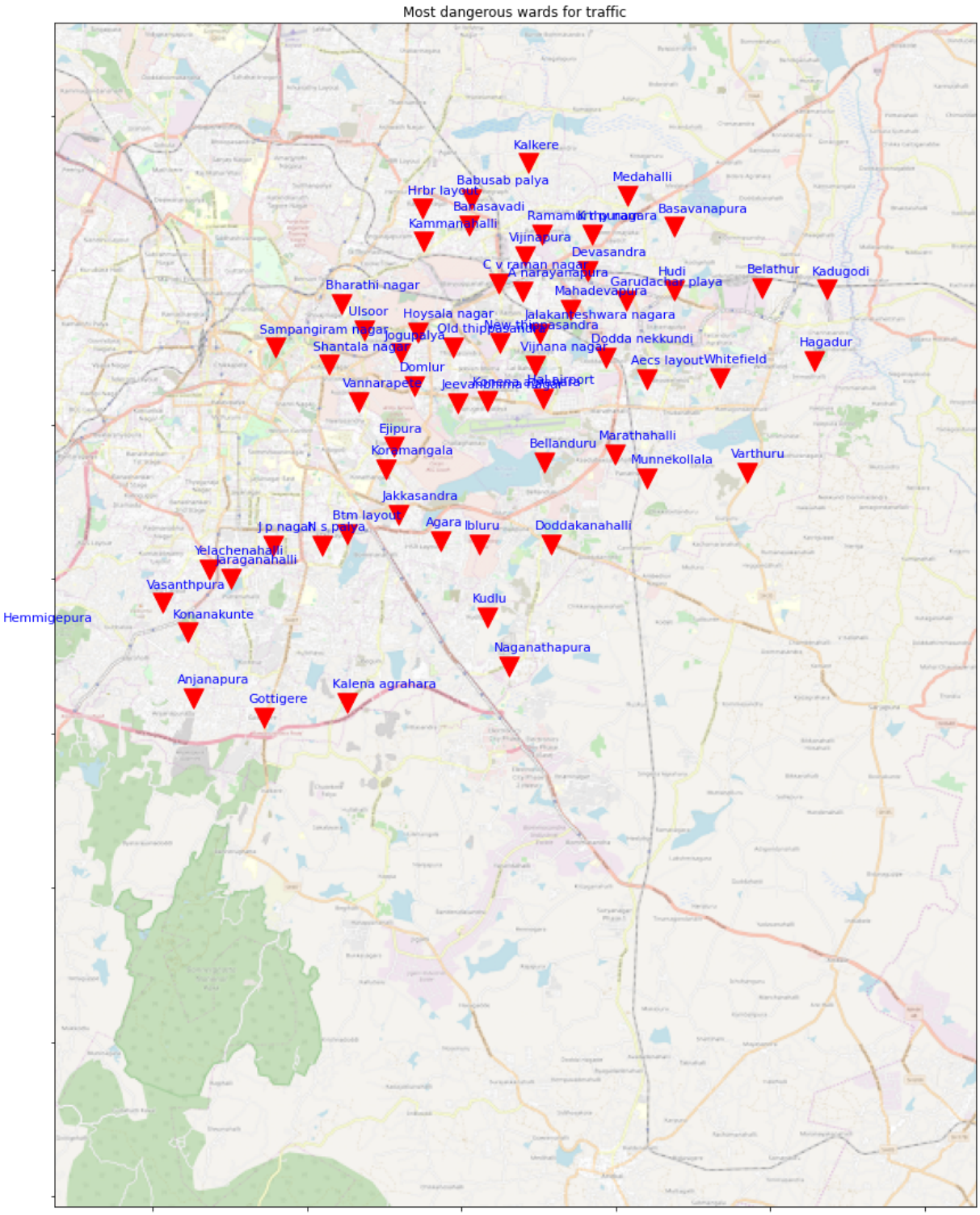
```
Whitefield                0.226385
Aecs layout               0.134578
Belathur                  0.134145
Hudi                      0.130369
Hagadur                   0.048852
Marathahalli              0.032507
Kadugodi                  0.031266
Garudachar playa          0.029525
Doddakanahalli            0.027614
Varthuru                  0.026642
Hal airport               0.026307
Basavanapura              0.020771
Bellanduru                0.016398
Dodda nekkundi            0.012858
Devasandra                0.010941
A narayanapura            0.008439
Jeevanbhima nagar         0.008406
Konena agrahara           0.007723
Munnekollala              0.006954
Vijinapura                0.006889
Vijnana nagar             0.006298
Old thippasandra          0.005694
Ibluru                    0.004984
C v raman nagar           0.004669
Mahadevapura              0.004032
Kudlu                     0.003763
New thippasandra          0.002981
Domlur                    0.002916
Jalakanteshwara nagara    0.002844
Naganathapura             0.001964
Jogupalya                 0.001701
Babusab palya             0.000978
Medahalli                 0.000919
Hoysala nagar             0.000690
Vasanthpura               0.000657
Hrbr layout               0.000624
Ramamurthy nagara         0.000604
Banasavadi                0.000460
Ulsoor                    0.000387
Kalkere                   0.000381
K r puram                 0.000374
Shantala nagar            0.000368
Konanakunte               0.000328
Anjanapura                0.000309
Hemmigepura               0.000296
Jakkasandra               0.000282
Koramangala               0.000210
Kalena agrahara           0.000190
N s palya                 0.000184
Vannarapete               0.000177
Btm layout                0.000171
Yelachenahalli            0.000164
Agara                     0.000164
Jaraganahalli             0.000158
Sampangiram nagar         0.000151
Gottigere                 0.000151
Kammanahalli              0.000112
Bharathi nagar            0.000099
Ejipura                   0.000092
```

```
J p nagar                    0.000092
Name: estimatedWardName, dtype: float64
```



In [66]:

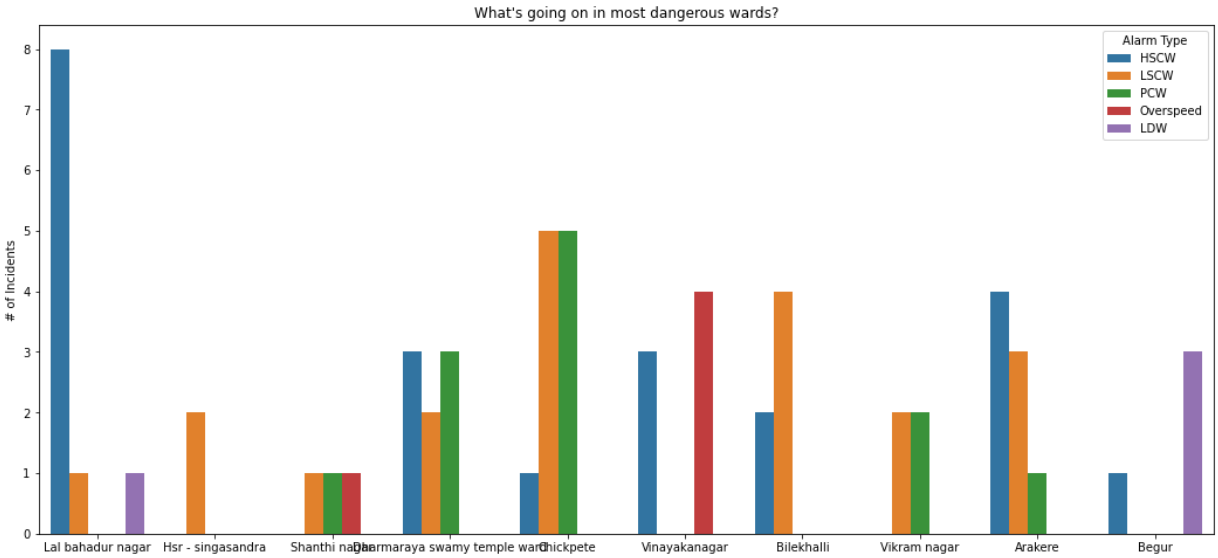
```
fig, axes = plt.subplots(figsize=(15,15))
cmap = plt.get_cmap("jet")

data = bbmp_data[bbmp_data.KGISWardName.isin(ward_top)]

axes.scatter(data.X, data.Y, marker="v", s=300,
                    c='red', zorder=1)
axes.set_title("Most dangerous wards for traffic")

# Plot Bangalore map image
epsilon = 0.01
bound_box = [lon_min + epsilon, lon_max + epsilon,
             lat_min + epsilon, lat_max + epsilon]

axes.imshow(bangalore_map, extent=bound_box, alpha=0.7, zorder=0)

# Add names of wards as text
for _idx, _ward_data in data.iterrows():
    axes.text(_ward_data.X  - epsilon/2, _ward_data.Y + epsilon/2,
              _ward_data.KGISWardName, color='blue', fontsize=11)

axes.set_yticklabels([])
axes.set_xticklabels([])

plt.tight_layout()
plt.show()
```



In [68]:

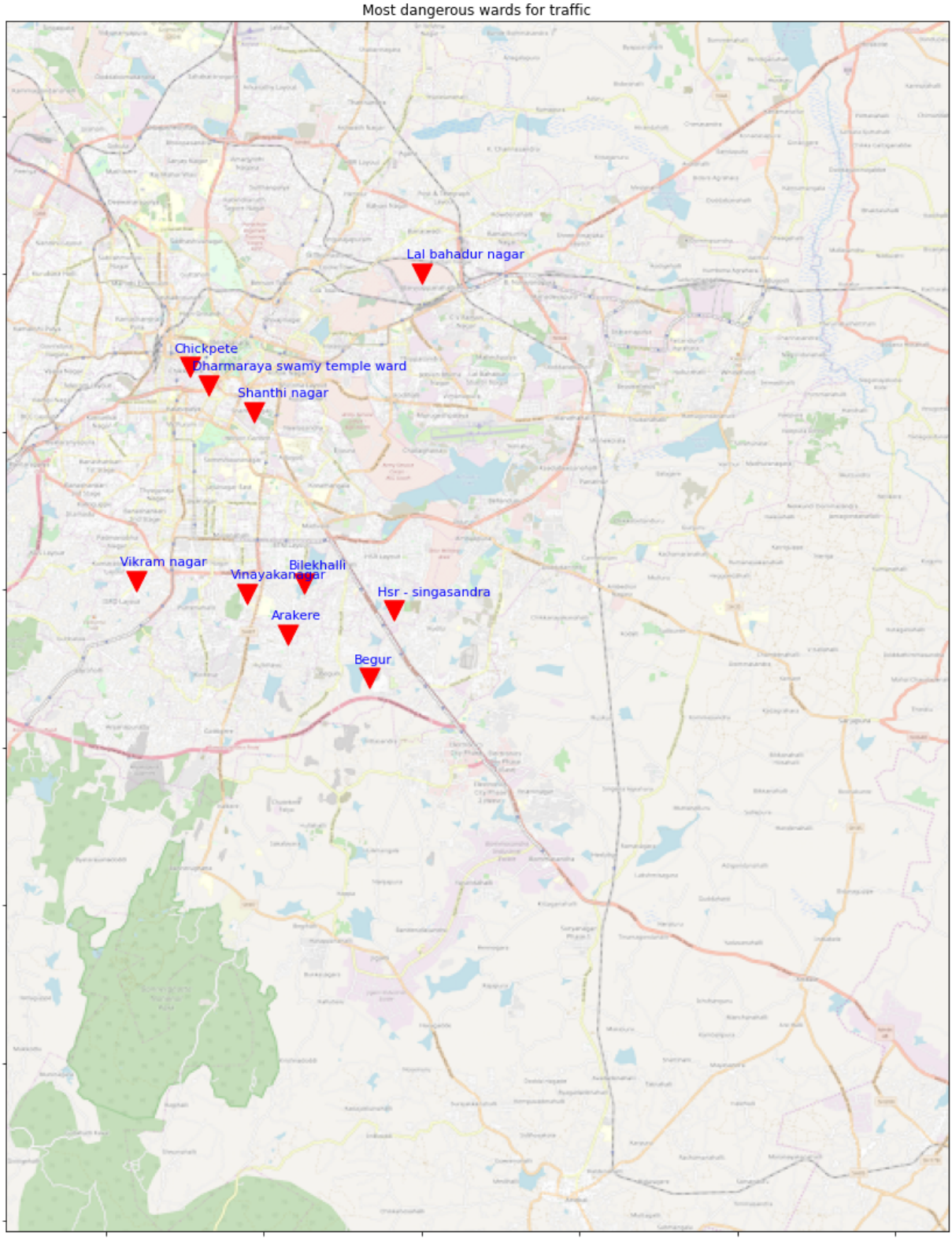
```
fig, axes = plt.subplots(figsize=(18,8))
data = df['estimatedWardName'].value_counts(normalize=True).sort_values()
data = data.head(10)
print(data)
ward_top = data.index

ward_top_data =df[df.estimatedWardName.isin(ward_top)]
sns.countplot(x='estimatedWardName', hue='alarmTypeCat', data=ward_top_data, ax=axes

axes.legend(title='Alarm Type')
axes.set_xlabel('')
axes.set_ylabel('# of Incidents')
axes.set_title('Safest ward')

plt.show()
```

```
Hsr - singasandra              0.000013
Shanthi nagar                  0.000020
Vikram nagar                   0.000026
Begur                          0.000026
Bilekhalli                     0.000039
Vinayakanagar                  0.000046
Arakere                        0.000053
Dharmaraya swamy temple ward   0.000053
Lal bahadur nagar              0.000066
Chickpete                      0.000072
Name: estimatedWardName, dtype: float64
```



In [69]:

```
fig, axes = plt.subplots(figsize=(15,15))
cmap = plt.get_cmap("jet")

data = bbmp_data[bbmp_data.KGISWardName.isin(ward_top)]

axes.scatter(data.X, data.Y, marker="v", s=300,
                   c='red', zorder=1)
axes.set_title("Most dangerous wards for traffic")

# Plot Bangalore map image
epsilon = 0.01
bound_box = [lon_min + epsilon, lon_max + epsilon,
             lat_min + epsilon, lat_max + epsilon]

axes.imshow(bangalore_map, extent=bound_box, alpha=0.7, zorder=0)

# Add names of wards as text
for _idx, _ward_data in data.iterrows():
    axes.text(_ward_data.X  - epsilon/2, _ward_data.Y + epsilon/2,
              _ward_data.KGISWardName, color='blue', fontsize=11)

axes.set_yticklabels([])
axes.set_xticklabels([])

plt.tight_layout()
plt.show()
```


Most dangerous wards for traffic
Lal bahadur nagar
Chickpete
Dharmaraya swamy temple ward
Shanthi nagar
Vikram nagar
Bilekhalli
Vinayakanagar
Hsr - singasandra
Arakere
Begur

In [ ]: