This Research acknowledges ND EPSCoR (UND0014140), the Office of RD&C (21418-4010-02294), and the UND Graduate School for the grant support.
OUTLINE OF THE PRESENTATION

- Introduction: Need for Situational Awareness of Smart-grid
- Proposed Situational Awareness Framework
- Development of User Interface for openPDC
- Data Visualization
- Data Clustering
  - DBSCAN Clustering
  - k-means Clustering
  - Multi-Tier k-means Clustering
- Results and Discussions
- Conclusion
# Need for Situational Awareness of Smart Grid

<table>
<thead>
<tr>
<th>Blackout Events</th>
<th>Affected Areas</th>
<th>Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 22, 2013 – Major ice-storm caused power failure.</td>
<td>Ontario to the maritime province in the far east and Michigan People affected – 1.1 million.</td>
<td>Ice storm</td>
</tr>
<tr>
<td>March 31, 2015 – Black-out, caused by technical failure, affected about 90% of Turkey.</td>
<td>90% of Turkey. People affected – 70 million.</td>
<td>Probable cyber attack.</td>
</tr>
</tbody>
</table>
INTEGRATED SOFTWARE SUITE (ISS)

Figure 1: Integrated Software Suite

Data Mining
* DBSCAN
* K-means
* 5-Tier K-means
Other algorithms

Data Visualization
* Unit Circle
* Geo-Spatial Mapping
* Box Whiskers

Intrusion Detection & Mitigation – Physical & Cyber
(MU 8000 Security Analyzer)

Alert Services
* E-mail
* Short Message

Forecasting
* Linear Regression
* Exponential Smoothing
* Holts Model
* ARIMA

VPN Server

Data Aggregator ex: PDC(openPDC)

Integrated Software Suite (MATLAB/Visual Studio/ C#)

Output Layer

Ordered SQL queries

Topology & State Estimator

Master Database (Oracle)
OpenPDC functions by receiving data broadcasted by a PMU and concentrating it, enabling archiving, rebroadcasting, and analysis of the phasor data. It provides around 30 samples per second.

**Functionalities:**
- E-mail Alarm
- Short Message Service alarm
- Location based monitoring

**Methodologies**
- C# used for all coding
- Visual Studio 2012 IDE used for development
- External libraries utilized:
  - Grid Solutions Framework
  - Google Static Maps API
  - .NET Framework 4.5
Subject: An alarm has triggered.
Time: 7/17/2014 3:33:58 PM
Name: TESTALARM
Threshold: 299300
Operation: Greater than or equal to Sever

An alarm has triggered.

To: Gellerman, Nickole
Time: 7/17/2014 5:37:04 AM
Name: TESTALARM
Threshold: 299300
Operation: Greater than or equal to Sever
Severity: Information
Description: Shelby Bus 1 + Voltage Magnitude

Data to monitor selected via dropdown box
Color scale a hue between red and blue, adjustable by user
Hovering over a data circle causes a pop-up with PMU information to appear

Figure 3: Short Message Service Alarm
Figure 4: E-mail Alarm
Figure 5: Location Based Monitoring System
DBSCAN CLUSTERING SCHEME

- DBSCAN is a density-based clustering algorithm that divides large regions with sufficiently high density into multiple clusters.
- DBSCAN considers two parameters as input excluding the data. They are $\varepsilon$ (Eps) and $MinPts$. $Minpts$ are the minimum number of points that are required to form a core, and $eps$ is the distance threshold from center of the cluster to its circumference of the cluster.

![Figure 6: DBSCAN Cluster Formation](image-url)
The k-means technique is a well-known and popular algorithm which was first proposed by Lloyd. Here, each cluster is represented by an adaptively changing centroid (also called a cluster center), starting from some initial values.

Figure 7: k-means Clustering
This paper presents a different version of k-means which we refer as multi-tier k-means clustering tailored for power system data sets.

The proposed approach dynamically forms clusters from 1 to 5 clusters depending on the data thresholds and fault type. They are: High Noise, High Border, Good Data, Low Border, and Low Noise points.

Capable of clearly distinguish the good, bad and the noisy data with the threshold inputs from the operator.

**Figure 8: Multi-tier k-means Cluster Formation**
Figure 9: Smart Grid Data Management Framework (SGDMF)
RESULTS AND DISCUSSIONS

- Data Visualization
  - Box Plot
  - Circle Representation

- Data Clustering
  - DBSCAN Clustering
  - k-means Clustering
  - Multi-Tier k-means Clustering
As phase angle varies between $-\pi$ to $+\pi$ (0 to 360 degrees) and the magnitudes are above 0 for the electric signals, unit circle representation is ideal smart-grid data.

The "Box Whiskers" is a statistical tool that allows observing a time-series data with minimum and maximum values in the series, standard deviations, mean and median values.
TEST SCENARIO: STEADY-STATE CONDITION

Figure 12: Clustering Schemes Applied on openPDC data under steady state condition
(a) DBSCAN, (b) k-means, (c) Multi-Tier
Figure 13: Clustering Schemes Applied on openPDC data under Heavy Load Conditions
(a) DBSCAN, (b) k-means, (c) Multi-Tier
TEST SCENARIO: LIGHT LOAD (LOW DEMAND) CONDITION

Figure 14: Clustering Schemes Applied on openPDC data under Light Load Conditions
(a) DBSCAN, (b) k-means, (c) Multi-Tier
Figure 15: Clustering Schemes Applied on openPDC data Under SLG Fault Conditions
(a) DBSCAN, (b) k-means, (c) Multi-Tier
DISTIBUTION OF DATA POINTS

<table>
<thead>
<tr>
<th>Load Condition</th>
<th>Noise Points (Red)</th>
<th>Border Points (Yellow)</th>
<th>Core Points (Green)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0.5</td>
<td>6.3</td>
<td>93.2</td>
</tr>
<tr>
<td>Heavy</td>
<td>0.078</td>
<td>8.96</td>
<td>90.5</td>
</tr>
<tr>
<td>Light</td>
<td>0.8</td>
<td>56.3</td>
<td>42.8</td>
</tr>
<tr>
<td>Fault</td>
<td>7.73</td>
<td>14.4</td>
<td>77.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load Condition</th>
<th>Cluster 1 (Blue)</th>
<th>Cluster 2 (Cyan)</th>
<th>Cluster 3 (Green)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>27.1</td>
<td>36</td>
<td>36.7</td>
</tr>
<tr>
<td>Heavy</td>
<td>25.3</td>
<td>40.1</td>
<td>34.4</td>
</tr>
<tr>
<td>Light</td>
<td>32.7</td>
<td>27.7</td>
<td>39.4</td>
</tr>
<tr>
<td>Fault</td>
<td>94.6</td>
<td>4.29</td>
<td>1.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load Condition</th>
<th>Low Noise (Blue)</th>
<th>Low Border (Cyan)</th>
<th>Good Points (Green)</th>
<th>High Border (Yellow)</th>
<th>High Noise (Red)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>10.53</td>
<td>89.47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Heavy</td>
<td>24.7</td>
<td>5.2</td>
<td>70.04</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Light</td>
<td>0</td>
<td>3.3</td>
<td>79.76</td>
<td>16.94</td>
<td>0</td>
</tr>
<tr>
<td>Fault</td>
<td>5.32</td>
<td>10.4</td>
<td>84.2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Steady-state condition: **Multi-tier** k-means performs best.
- Heavy-load condition: **DBSCAN** performs best.
- Light-load condition: **DBSCAN** performs best.
- Fault condition: **Multi-tier** performs the best.
CONCLUSION

- An Integrated Software Suite (ISS) has been developed to apply decision-making data-mining algorithms on time-synchronized synchrophasor data.

- A novel, Multi-Tier variation of the k-means algorithm is presented, and its performance metrics are studied against common clustering techniques to classify and detect bad data, event detection, and alarm service applications.

- A comparative analysis has been carried out between the three data clustering algorithms, DBSCAN, k-means and the Multi-Tier k-means.

- It is believed that such a framework will enable the grid’s system operators to utilize novel algorithms in order to enhance situational awareness about the grid. The framework is scalable and suitable for streaming time-series data sets.
FUTURE WORK

- Study application of forecasting algorithms like:
  - Time Series Data Analysis
  - Linear Regression
  - Exponential Smoothing
  - Holt’s Model
- Topology based State Estimator
- Intrusion Detection and Mitigation Systems
REFERENCES:


[19]
THANK YOU…

Questions???
**K MEANS CLUSTERING SCHEME**

**Iteration – 1**
Centroids chosen in random: 
\( C_1, C_2, C_3 \)

Calculate new centroids based on data points in each cluster:
\( C_1' \neq C_1, C_2' \neq C_2, \\
C_3' = C_3 \)

**Distance Metric used: Euclidean**
\[
D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

**Iteration – 2**
Centroids used from previous: 
\( C_1', C_2', C_3' \)

Calculate new centroids based on data points in each cluster:
\( C_1'' = C_1', C_2'' \neq C_2', \\
C_3'' = C_3' \)

**Iteration – 3**
Centroids used from previous: 
\( C_1'', C_2'', C_3'' \)

Calculate new centroids based on data points in each cluster:
\( C_1''' = C_1'', C_2''' = C_2'', \\
C_3''' = C_3'' \)

Figure: k-means Cluster formation
Inputs for the Algorithm

\[ X = \text{Dataset} \]
\[ Eps = \text{Min. distance between two points} \]
\[ D = \text{Min. number of points required to make core} \]

Distance Metric used: Euclidean

\[ D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \]
<table>
<thead>
<tr>
<th>Stage - 1</th>
<th>Type - I: Normal k-means algorithm is applied with 3 clusters and pre-defined values of centroids $C_1$, $C_2$, $C_3$.</th>
<th>Running iterations until the dataset is clustered into three groups or less.</th>
<th>Based on the transmission line data, the three clusters are then modified to generate potentially two more clusters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage - 2</td>
<td>Type - I: Thresholds are setup to divide the dataset into five different cluster regions.</td>
<td>The regions are given a separate color to indicate the level of contingency: high noise (red), high border (yellow), core (green), low border (cyan), and low noise (blue).</td>
<td></td>
</tr>
<tr>
<td>Stage - 1</td>
<td>Type - II: Normal k-means algorithm is applied with 3 clusters and pre-defined values of centroids $C_1$, $C_2$, $C_3$.</td>
<td>Running iterations until the dataset is clustered into three groups or less.</td>
<td>Based on the transmission line data, the three clusters are then modified to generate potentially two more clusters.</td>
</tr>
<tr>
<td>Stage - 2</td>
<td>Type - II: Thresholds are setup to divide the dataset into five different cluster regions.</td>
<td>The regions are given a separate color to indicate the level of contingency: high noise (red), high border (yellow), core (green), low border (cyan), and low noise (blue).</td>
<td></td>
</tr>
</tbody>
</table>

**Inputs for the Algorithm**
- $X$ = Dataset
- $V$ = Expected voltage of Transmission line
- $S$ = Allowable range for the line voltage to fluctuate

**Distance Metric used: Euclidean**

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$