Emergency Medical Services Demand Forecasting: A Survey of Machine Learning Methods

Richard J. Martin, Fakhri Abbas, David Vutetakis, Archit Parnami

ITCS 8156 Machine Learning – Spring 2017 Course Project

The University of North Carolina at Charlotte

College of Computing & Informatics

Abstract

Emergency medical services (EMS) are responsible for providing medical care to populations within defined geographic regions which includes transportation to local facilities such as emergency departments and hospitals. Research has shown a direct relationship between the EMS response time and the patient's survival rate, which emphasizes the importance of minimizing the response time of each individual call request. Reliable forecasting of emergency call volumes within a geographic region can improve demand planning, staffing and resource allocation to reduce mortality rates. Previous researchers have attempted to model the demand trends for future predictions through techniques such as moving averages and artificial neural networks (ANN), but a significant gap in the literature exists for novel forecasting methods capable of the resolution and accuracy required by EMS agencies. In this study, two novel methods of dynamic spatial segmentation were implemented using self-organizing maps and k-means clustering. A training dataset consisted of the individual call volumes of each cluster binned by one-hour time blocks. A survey of supervised machine learning methods was analyzed, which included Neural networks, Naïve Bayes, Decision Trees with AdaBoost, K-Nearest Neighbors, and Support Vector Machines. A base model was created using a neural network approach and used to compare the performance of each of additional algorithms measured in terms of accuracy and F-Score. The results showed a significant class imbalance which needed to be overcome, as well as the importance of feature selection. The support vector machine produced the best results due to its high overall accuracy in addition the relative distribution of individual prediction accuracy between call volume classes.

Keywords: emergency medical services, machine learning, neural network, support vector machine, k-nearest neighbor, adaboost decision tree, naïve bayes

I. Introduction

Emergency medical services (EMS), commonly referred to as ambulance, paramedic or prehospital emergency services, are a critical component in the delivery of urgent medical care to communities. EMS agencies are organizations charged with the responsibility of providing out-of-hospital acute medical care to the population of a defined geographic area such as a city, county or local municipality. EMS agencies also provide transportation to local clinical care facilities, such as hospitals and emergency departments, for patients who are unable to transport themselves due to the nature of their condition or circumstances. Depending on the local healthcare infrastructure, EMS agencies may be owned and operated by local governments, healthcare systems or private organizations. Additionally, EMS ambulances and staff may provide transportation services to and from local or non-local clinical care facilities. The primary goal of EMS agencies is to minimize their response time to individual call requests, and in doing so, minimize the rate of mortality and morbidity [1].

By their very nature, EMS systems are extraordinarily complex. The demand for ambulances is dynamic and is known to fluctuate spatially and temporally based on the time of day and day of the week [2]. EMS managers and dispatchers are faced with the evolving task of deploying the ambulances and personnel required to provide adequate coverage for a defined geographic service area given limited resources. Dispatchers have the option of re-deploying their fleet of ambulances to compensate for spatiotemporal demand fluctuations, but the scope of these adjustments is restricted by the predetermined staffing plans for a given period. Industry and academic researchers have conducted various studies focused on developing novel deployment strategies, and associated staffing plans, in an effort to reduce response time variability and maximize service coverage [3]. These deployment models, developed based on historical data, are ultimately dependent on detailed call demand forecasts. Related research has sought to identify more sophisticated call forecasting approaches to improve the predictive models used for demand planning.

Many of these studies focused on developing forecasts for broader time periods and geographic areas. While these types of forecasts are valuable for strategic and tactical capacity planning over longer periods of time (i.e. monthly or yearly), they are not very useful for short-term operational

decisions, such as daily and hourly deployments. Currently, there are few forecasting models that attempt to predict ambulance call volumes at different spatiotemporal granularities. This represents a significant gap in the literature, as researchers have not developed novel approaches that produce forecasts at the level of detail required to be used by EMS agencies for short-term deployment planning. An additional major gap in the current research is investigations that explore the application of machine learning algorithms to predict EMS call demand. To date, only two studies have been conducted that applied machine learning to this domain. Specifically, Setzler et al. [4] and Chen et al. [5] developed implementations of Artificial Neural Networks that yielded marginal results compared to current industry practices.

The focus of this study will be to produce call demand forecasts at various scales of time and space using a survey of different machine learning algorithms. Our current plan is to explore various configurations of Neural Networks as well as applications of Naïve Bayes, Support Vector Machines, K-Nearest Neighbors and AdaBoost on Decisions Trees. We will begin our analysis by implementing an Artificial Neural Network based on the original paper by Setzler et al. [4]. This initial neural network will be used as our base model for comparison and advancement. As proposed by Setzler et al. [4], our forecast resolutions will be determined by a combination of hourly time-frames and spatial clusters for a specific geographic area. These different resolutions will be represented by coding our input/output features to different configurations. Our primary objective is to conduct an exhaustive study, comparing the performance of various machine learning algorithm implementations, and ultimately identify approaches that are suitable for predicting future demand for emergency medical services. Along the way, our hope is that we can outperform the current state of the art based on the current literature. This study and our subsequent results will contribute to the overall emergency medical services research community.

II. Literature Review

As noted by Aringhieri et al. [1], Channouf et al. [2], and McConnel & Wilson [6], some of the earliest works related to emergency medical services demand forecasting appeared in the 1970's. The first three prominent investigations were conducted by Aldrich et al. [7], Kvlseth & Deems [8], and Siler [9]. In each of these studies the researches performed regression analyses against

EMS call volume data joined with socio-economic demographic data to develop causal forecasting models. Following in the footsteps of their predecessors McConnel & Wilson [6] used demographic and call volume data to perform a study in 1998 using a variety of statistical testing methodologies, including Chi-Squared and Tukey's Range Test, to examine the impact of the aging society on emergency medical services demand. Nearly a decade later in 2007, Channouf et al. [2] performed a study concentrated on generating daily and hourly EMS call volume forecasts by applying a variety time-series models. Emphasizing the importance of accurately predicting demand, Channouf et al. highlight the fact that reliable forecasts serve as essential inputs into EMS planning models and accompanying staffing plans. Channouf et al. focused exclusively on time as the central variable. In preparing their data for analysis, records were aggregated based on the number of calls occurring during each hour of the day. While previous studies produced forecasts for broader time periods, such as months or years, Channouf et al. were the first to recognize that the demand for emergency medical services varies significantly based on the time of the day and the day of the week [2], and therefore took a novel approach to producing forecasts at the daily and hourly level. In each of the preceding investigations the researchers focused on time as the single dimension.

In 2009, Setzler et al. [4] performed a significantly novel study aimed at producing EMS call volume forecasts at various spatiotemporal granularities using artificial neural networks. The data used to conduct their investigation consisted of emergency calls dispatched between 2002 and 2004 by MEDIC, an EMS agency responsible for serving the populace of Mecklenburg County, North Carolina. While Channouf et al. were the first to formally state that the demand for ambulances changes significantly based on the time of the day and the day of the week [2], Setzler et al. expanded on this observation by incorporating a spatial component of demand into their forecasting methodology. Individually each call record contained information related to the time of the call and latitude-longitude coordinates identifying the call location. This enabled the researchers to aggregate the call data at various gradations of time and space. Specifically, they grouped the call volumes into 1-hour and 3-hour time range buckets, and geographically into 2-mile x 2-mile and 4-mile x 4-mile square mile grid blocks; creating a total of four different model configurations at different levels of specificity. It is important to note that the researchers selected these different configurations arbitrarily, and not based on any quantitative

measurements or observations. Alternatively, they could have used various clustering and feature mapping approaches such as k-means and self-organizing maps. Using the prominent geographic information system (GIS) platform ArcGIS, 2-mile x 2-mile and 4-mile x 4-mile grid layouts were layered over a map of Mecklenburg Country creating a total of 168 and 40 grid blocks, respectively. Citing that many of the previous studies leveraged traditional casual forecasting approaches, Setzler et al. postulated that artificial neural networks (ANN) were a viable alternative as they do not require specific assumptions about the data or error terms, are able to adapt to complex data sets and patterns, and can learn and model both linear and non-linear relationships [5]. They designed their neural network based on the spatial grid layout where (n) represented the total number of grid blocks, either 168 or 40. They also used four different input variables, (H) which represented the hourly time bucket (24 or 8 in total), (S) for the season of the year (1-4), (D) for the day of the week (1-7), and (M) representing the month of the year (1-12). Each grid block (n) had 4 input nodes into the ANN correlating to each of the four input variables (H, S, D, M). The output from the ANN represented the call volume forecasts for each of the (n) grid block locations, for a total of 168 or 40 forecast values depending on the configuration. To benchmark the performance of their model, Setzler et al. compared the forecasts produced by their ANN against forecasts for each grid block produced using a method applied by MEDIC and others in the EMS industry, a 20-point moving average [Setzler]. Their results showed that the ANN approach moderately outperformed the moving average forecasts at the 4 x 4 mile (40 grid) 1-hour and 3-hour granularity levels. However, at the 2 x 2 mile 1-hour scale the moving average forecasts were more accurate on average over the ANN approach and exhibited no statistically significant difference at the 2 x 2 mile 3-hour level. The authors ultimately concluded that the performance and simplicity of the moving average method currently in use by MEDIC and other EMS agencies suggest no reasonable justification for implementing an artificial neural network for demand forecasting. Despite their results, Setzler et al. [4] were pioneers in exploring the application of artificial neural networks for forecasting emergency medical services demand.

III. Data Preparation & Establishing Base Model

To conduct this study, we used the same EMS call volume data as Setzler et al. The complete dataset consists of all the emergency calls dispatched by MEDIC, the Mecklenburg County EMS

Agency, between 1997 and 2004. As discussed in our literature review, Setzler et al. focused on four temporal input features; season, month, day of the week, and hour. They also coded each of the call records into geographic grids (4x4 mile and 2x2 mile grids) based on the latitude and longitude coordinates of the original call locations. As previously noted, these grid sizes were selected arbitrarily by the researchers. For the purposes of establishing our base model we used the same dataset and similar configurations. However, we elected to code the call records into different geographic clusters using both a k-means and self-organizing map (SOM) approach. In considering the number of clusters, we referred to the findings by Setzler et al. They found that while using a 4x4 mile (40 clusters) and 2x2 mile (168 clusters) grid configuration, increasing the level of granularity resulted in a significant decrease in the forecast accuracy. Therefore, we decided on an initial clustering size of k=36 for both k-means and SOM. Our k-means clustering implementation was developed in Python using the popular scientific computing package SciKit-Learn. For the SOM implementation, we wrote a custom MatLab implementation. Visualizations for both the SOM and k-means outputs are provided below in Figure 1 and 2 respectively.

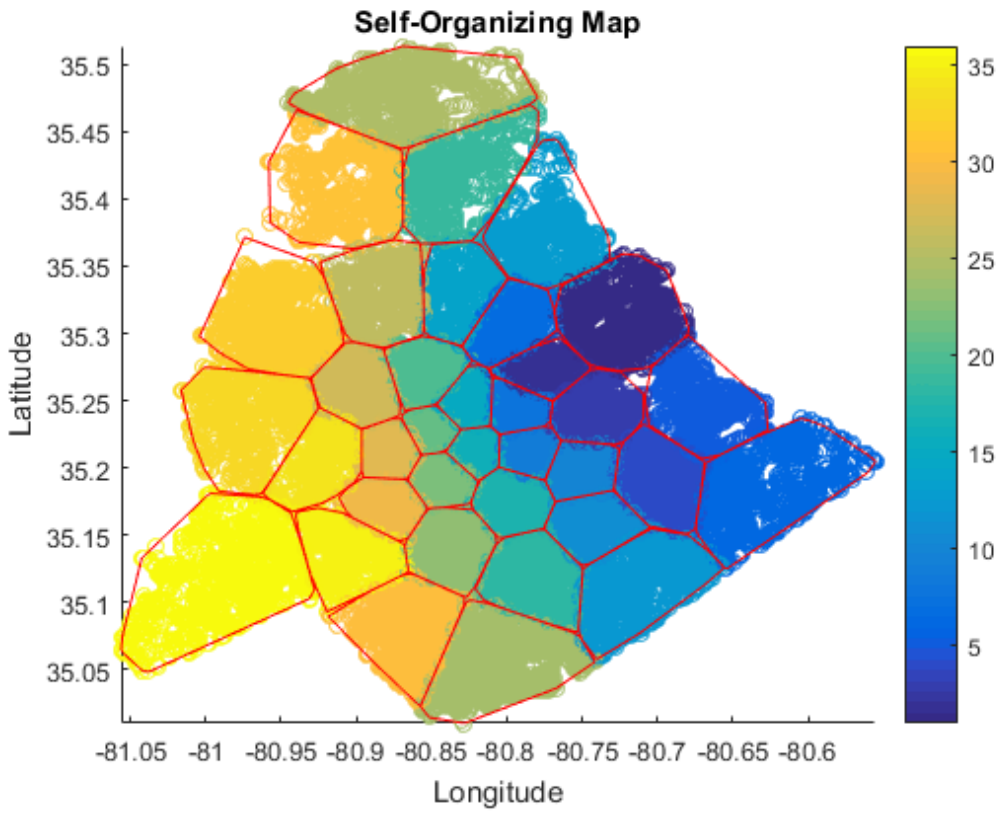

Figure 1: MEDIC EMS Call Locations Self-Organizing Map

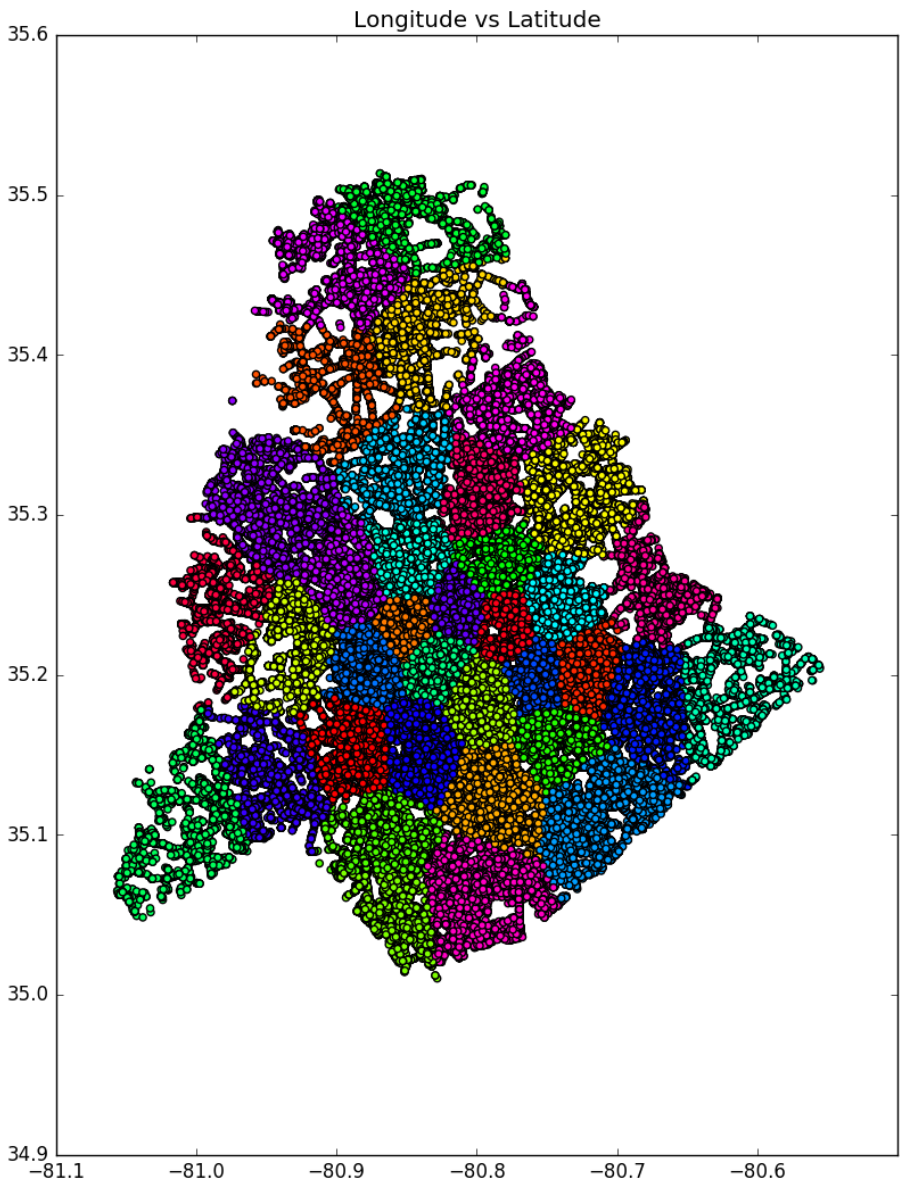

Figure 2: MEDIC EMS Call Locations K-Means Clusters

Using the original MEDIC data, we processed all the records through both the SOM and k-means clustering models to produce our geographic location coded training and testing datasets.
To ensure uniformity with the original study, we filtered the data to only include call records from 2002-2003 for training and 2004 call records for testing. Additionally, we coded all the temporal features to exactly match the coding schema of the original study and filtered out all features except month, day of the week, hour, season, cluster location and call volume. The final state model developed by Setzler et al [4] to predict call volume at a given cluster location was a multilayer feedforward neural network with backpropagation learning, configured using one

hidden layer with one node for each input, and applied the sigmoid activation function. For our base model, a neural network with identical configurations was implemented in Python using SciKit-Learn. The base model code and corresponding dataset files are included with our submission in the "Base Model Neural Network" folder. The results from the base model run are shown below for both the k-means and SOM datasets.

```
K-Means Clusters
Training Accuracy: 0.662547534492
Cross Validations Mean Accuracy: 0.662779705335
Test Accuracy: 0.665941307368
Mean Squared Error: 0.944319
Root Mean Squared Error:0.971761

Self-Organizing Map Clusters
Training Accuracy: 0.682183600993
Cross Validation Mean Accuracy: 0.68131442651
Test Accuracy: 0.679710839819
Mean Squared Error: 0.769888
Root Mean Squared Error:0.877433
```

Setzler et al. stated that at a (4 x 4) mile granularity using 1-hour intervals, which is a similar level of granularity to our clustering approach, their model produced accurate forecasts with 0.25 error 75-76% of the time. Given the fact that our base model is configured to produce discrete integer value predictions, versus continuous values, and is implemented using a novel clustering strategy a test accuracy of approximately 67% is reasonably comparable. Comparing the test accuracies and mean squared error, which is arguably one of the most important indicators of prediction performance, the self-organizing map cluster coding outperforms the k-means clustering approach on the base model.

IV. Additional Feature Selection & Transformation

Exploring the performance of the base model further we assessed the distribution of the predicted values. We found that the base model was actually only predicting (1's) for all of the test records. For example, using SOM the base model predicted 81,339 instances of '1' for call volume. The test set only contains 81,339 records, and given that the original test set has a total of 55,287 records labeled as having call volumes of '1' the current accuracy of the base model is simply the ratio 55,287/81,339 = 67.97%. Our initial reaction to this finding was that a class imbalance present in the data was the root cause of the issue. This finding is also consistent with

a similar drawback encountered by Setzler et al. They found that a model that simply forecasted all zeros for each grid location every hour produced less error than any prediction model they developed [4]. This is a problem inherit to the nature of EMS calls. Predicting the total number of calls for an entire geographic region using broader time periods such as a day, week, or month is a relatively simple problem. The complexity is derived from finer levels of granularity. Figure 3 illustrates an example of a possible distribution of volumes for a given period. At finer granularities, the number of instances for '0' or '1' calls far outweighs the number of instances for call volumes greater than '1'.

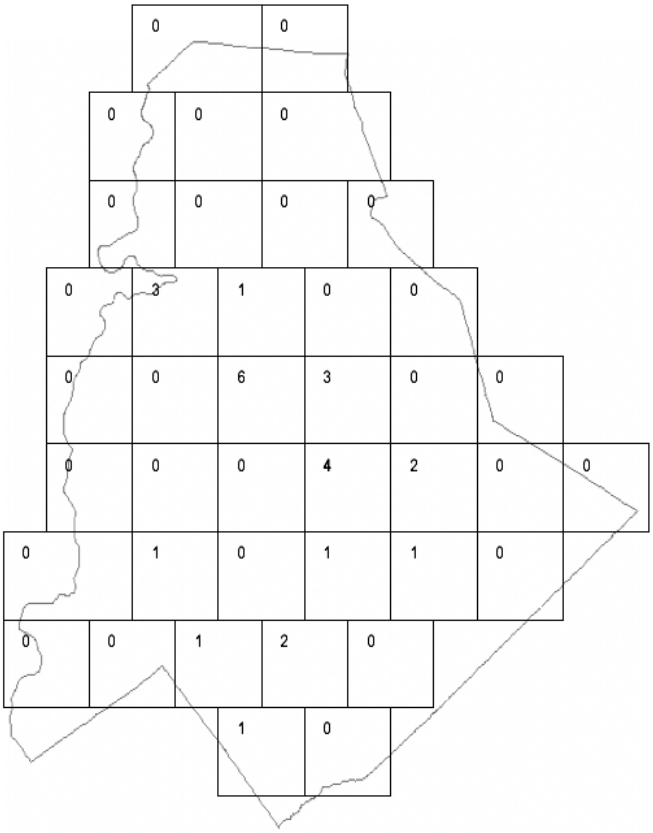

Figure 3: Example of Possible Call Distribution for 1-Hour Period

This finding was also an indicator that using a traditional prediction accuracy measurement is not a good judgement of model performance. Therefore, we explored using F-Score as an alternative performance measurement. The F-score metric, or F1 score, can be used to measure a model's accuracy based on the relationship between true and false predictions for individual labels. Essentially, the F-score combines the precision and recall in one equation to capture the accuracy. Precision measures the ratio between true-positives to true-positive and false-positive.

While, recall measures the ratio between true-positives to true-positives and false-negatives. Precision, recall and f-score are defined by the following equations.

$$Precision = True\ Positive / (True\ Positive + False\ Positive)$$
$$Recall = True\ Positive / (True\ Positive + False\ Negative)$$
$$F\text{-}Score = 2 * (Precision * Recall) / (Precision + Recall)$$

To remedy the class imbalance problem, we revisited the original training data and analyzed the current label distribution. As an example, the distribution for the self-organizing map training dataset is shown below in Table 1.

Table 1: Distribution of Labels in Self-Organzing Map Training Dataset

Label	Count	Percent	Running % Sum
1	108532	68.22%	100.00%
2	36782	23.12%	31.79%
3	9807	6.16%	8.67%
4	2860	1.80%	2.51%
5	855	0.54%	0.71%
6	192	0.12%	0.17%
7	42	0.03%	0.05%
8	16	0.01%	0.02%
9	9	0.01%	0.01%
Total	159095	100%	-

With the last six labels (4-9) constituted a total of 2.51% of the total records we consolidated them all into one label for volumes 4+ to reduce the total number of labels from nine to four (i.e. 1, 2, 3, 4+). Another observation we made from the dataset is that currently there are no records for instances of zero call volumes. This is unrealistic, and incorrectly trains any given model to assume there will always be a call instance no matter what feature values are given. Therefore, we wrote a custom python script to generate call records with zero volume for every combination

of time and location that didn't have a call instance. We then appended these records to our original datasets.

V. Supervised Learning Implementations & Results

ii. Neural Network

Expanding on the base model, we continued our primary analysis by performing a collection of iterations with the SciKit learn implementation of a neural network using both the SOM and k-means datasets. The model was evaluated using the two sets of features shown below in table 2. The first set of features we used were identical to the features used in the base model, while the second set contains a slightly different collection of features that are believed to capture the behavior of the data more accurately.

Table 2: Neural Network Feature Selection

Set	Month	Hour	Day	Day of Week	Season	Week Number	Year	Cluster Code
First	✓	✓		✓	✓			✓
Second	✓	✓	✓	✓				✓

Table 2

For each new iteration, various parameters were adjusted to observe their impact. Specifically, we focused on changing the activation function and the number of nodes in the hidden layer. Table 3 below specifies the selected combinations used in each iteration. A total of 32 different runs were performed for the different combinations.

Table 3: Neural Network Paramater Combinations

Category	Values
Hidden Layer Nodes	3,4,5,6,7,8,9,10
Activation Function	'identity', 'logistic', 'tanh', 'relu'

After analyzing the initial results, the volume predictions generated by the k-means data slightly outperformed the SOM dataset. Among all the runs, the highest F-Score achieved was 0.67. Focusing on the F-Score result for each class label as the primary performance metric indicated which model configuration performed the best. Results are shown below in Table 4 and Figure 4.

Table 4: Neural Network Iterations Top Results

Clustering	Nodes	Activation	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	F-score "label-4"	F-score	Accuracy
Kmean	**7**	**Logistic**	**0.87**	**0.07**	**0**	**0**	**0**	**0.67**	**76%**
Kmean	3	Tanh	0.86	0	0	0	0	0.65	75%

The first model shown in the above table which used the k-means dataset, (7) hidden nodes, and the logistic activation function outperformed all the other iterations. Its accuracy is the best among all different runs and is the only run that predicts values other than '0'. To validate the model, cross validations was performed using 10 folds, each fold training with 70% of the data and 30% for testing. The mean accuracy for all folds was 76%, which matches the model's testing accuracy.

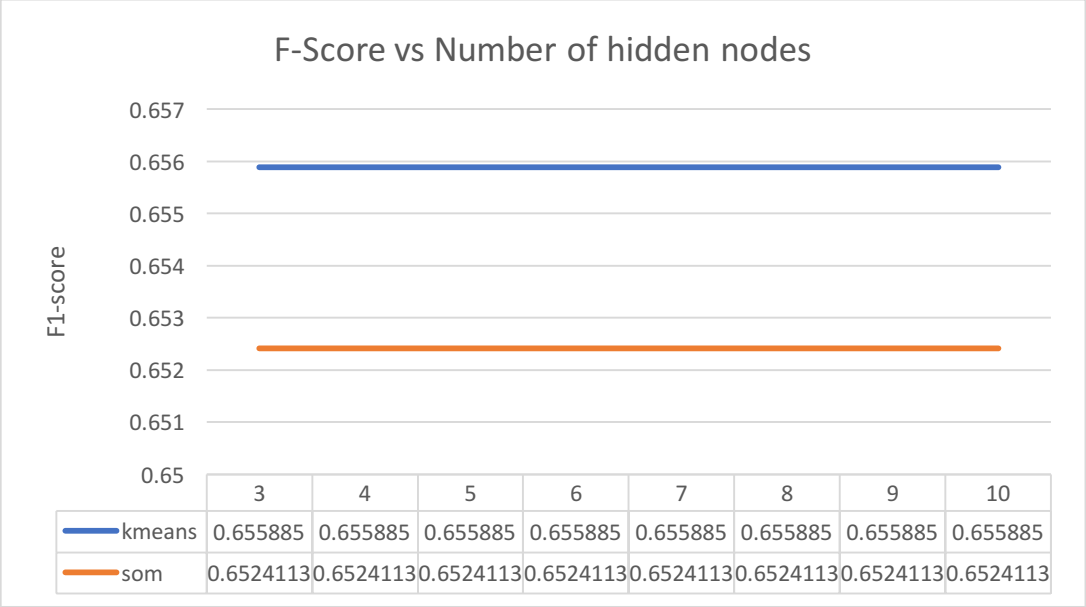

Figure 4: Neural Network F-Score Results for K-Mean and SOM using Identity Function

The selected model was also tested using the feature set from the original base model against both the k-means and SOM dataset. Performance declined for both, indicating that the second set of features generalizes better.

Table 5: Neural Network Performance Using Base Model Features Set

Clustering	Nodes	Activation	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	F-score "label-4"	F-score	Accuracy
Kmean	7	logistic	0.86	0	0	0	0	66%	76%
som	7	Logistic	0.86	0	0	0	0	65%	75%

iii. Naïve Bayes

Moving on to the next implementation we applied Naïve Bayes, which compared to the complexity of neural networks is a relatively simple technique for constructing classifiers. To remain consistent with the previous neural network implementation we ran the model using both the SOM and k-means datasets. The model was evaluated using the three different feature sets noted below in table 6. The first set of features used were identical to the features used in the base model, while the second and third sets contain a slightly different collection of features that are believed to capture the behavior of the data more accurately.

Table 6: Naïve Bayes Feature Selection

Set	Month	Hour	Day	Day of Week	Season	Week Number	Year	Cluster Code
1(Base)	✓	✓		✓	✓			✓
2	✓	✓	✓	✓				✓
3	✓	✓	✓					✓

Unlike other supervised learning methods, Naïve Bayes models typically do not have a collection of parameters that can be tweaked and tuned. Therefore, to perform a comprehensive analysis we explored the performance of both datasets across all three features sets using multiple iterations. We also returned to the original datasets that did not include the zero volume records to explore any possible improvements to prediction accuracy. Table 7 below summarizes the results for each run. We have labeled the dataset that does not include zero volume records as dataset 1, and the other as dataset 2

Table 7: Naïve Bayes Iterations Top Results

Dataset	Cluster Type	Feature Set	Duplicates Allowed	F – 0	F – 1	F – 2	F- 3	F- 4	F Score (%)	Accuracy (%)
1	SOM	1	Yes	-	0.80	0.0002	0.0	0.0	55	67.95
1	SOM	2	Yes	-	0.80	0.0004	0.0	0.0	55	67.95
1	SOM	3	Yes	-	0.80	0.0004	0.0	0.0	55	67.94
1	K-Means	1	Yes	-	0.78	0.07	0.003	0.0	53.8	64.56
1	K-Means	2	Yes	-	0.78	0.07	0.0	0.0	53.9	64.65
1	K-Means	3	Yes	-	0.78	0.07	0.0	0.0	53.9	64.61
2	SOM	1	Yes	0.84	0.023	0.00	0.0	0.0	63.8	73.85
2	SOM	2	Yes	0.85	0.21	0.004	0.0	0.0	63.8	73.96
2	**SOM**	**3**	**Yes**	**0.84**	**0.024**	**0.0004**	**0.0**	**0.0**	**63.9**	**73.89**
2	**K-Means**	**1**	**Yes**	**0.80**	**0.173**	**0.053**	**0.003**	**0.0**	**63.86**	**67.42**
2	K-Means	2	Yes	0.81	0.16	0.05	0.0	0.0	63.8	67.86
2	K-Means	3	Yes	0.81	0.16	0.05	0.0	0.0	63.8	67.87

From the results above it is evident that the highest accuracy and f-score achieved is 63.9 and 63.86 respectively. Taking a closer look at individual f-scores of labels we can observe that these high scores are primarily influenced by just one or two labels. That is the model is mostly predicting same labels. Due to high number of '1' labels in dataset 1, the model mostly predicted 1's. As previously noted, to remedy this problem we introduced '0' labels to incorporate calls which did to happen. This resulted in a huge influx of zero labels. The probability of '0' labels naturally increased and the model adapted to predicting zeros instead of 1's. Overall SOM achieved slightly better results than K-means clustering.

iv. Decision Tree w/AdaBoost

Continuing our analysis, we implemented AdaBoost using a series of small decision trees of varying sizes as weak leaners. A collection of iterations was performed using different sets of parameters and total number of estimators to accomplish the prediction task. The model was evaluated using the two sets of features shown below in table 8. The first set of features is identical to the features used in the base model, while the second set contains a slightly different collection of features that are believed to capture the behavior of the data more accurately.

Table 8: Decision Tree w/ AdaBoost Feature Selection

Set	Month	Hour	Day	Day of Week	Season	Week Number	Year	Cluster Code
First (Base)	✓	✓		✓	✓			✓
Second	✓	✓	✓	✓				✓

The different combinations of model parameters used are shown below in Table 9. Primarily, the model was tuned by adjusting the tree depth and number of estimator parameters. Additionally, each unique model was trained and evaluated using the k-means and SOM clustering label datasets.

Table 9: Decision Tree Parameter Combinations

Category	Values
Tree Depth	2,3,4,5,6,7,8,9
Number of estimators	50,100,150,200,250,300

A total of 48 runs representing the different combinations of tree depth and number of estimators were performed. After analyzing the initial results, the volume predicts generated by the k-means data slightly outperformed the SOM dataset. Among all the runs, the highest F-Score achieved was 0.69. Focusing on the F-Score result for each class label as the primary performance metric indicates which model configuration performed the best. Results are shown below in Table 10 and Figure 5.

Table 10: Decision Tree w/ AdaBoost Iterations Top Results

Clustering	Tree Depth	Number of estimators	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	F-score "label-4"	F-score	Accuracy
k-mean	2	50	0.88	0.07	0.07	0.08	0.37	69%	77%
k-mean	9	200	0.66	0.25	0.16	0.45	0.98	57%	52%
k-mean	**9**	**250**	**0.69**	**0.25**	**0.17**	**0.45**	**0.99**	**59%**	**54%**

The first model achieved the highest accuracy, but again only because it mostly predicted (0's) due to the class imbalance, while the second and third model performed much better based on the distribution of f-scores across all the label values. With a tree depth of 9 and 250 estimators the

third model configuration has the best overall prediction accuracy across all class labels. To validate this final model, cross validation was conducted using 10 folds with 70% of the dataset allocated for training and 30% for validation. The mean training accuracy for all the folds was 53% which is a 1% less than the models testing accuracy.

Figure 5: AdaBoost F-Score Results for K-Mean and SOM with Tree Depth of 2

The selected model was also tested using the feature set from the original base model against both the k-means and SOM dataset. Performance declined for both, indicated that the second set of features generalizes better.

Table 11: AdaBoost Performance Using Base Model Features Set

Clustering	Tree Depth	Number of estimators	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	F-score "label-4"	F-score	Accuracy
Kmean	9	250	0.55	0.21	0.11	0.13	0.49	46%	38%
som	9	250	0.53	0.21	0.11	0.13	0.43	45%	38%

v. K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm was implemented using various K-values and with additional feature selection and sub-sampling. To reduce the effects of the class imbalance, the number of observations within each class was adjusted through random sub-sampling. This was performed by first determining the number of samples with the highest volumes (three or higher), then scaling down the number of smaller values (0, 1 & 2) proportionally.

The next step in the analysis was selection of training features. Many of the features conveyed redundant information, and introduced more noise than useful information due to the seasonality of the data. For this model, the best time components used for training were the Month, Day, and hour.

Table 12: KNN Feature Selection

Set	Month	Hour	Day	Day of Week	Season	Week Number	Year	Cluster Code
1(Base)	✓	✓		✓	✓			✓
2	✓	✓	✓	✓				✓
3	✓	✓	✓					✓

The KNN model was trained using many combinations of sampling, feature selection, and other parameters. Training using the entire dataset produced a high apparent accuracy, but was predicting zero volume in all instances similar to the other models. After class balancing, the training features of set three produced the best results. The results showed that using a single month's data generalized well to the other months, reducing the amount of data required for training. K-values from 1 through 10 were tested with the best results obtained using k=1 nearest neighbors. An accuracy of 74.1% was achieved, but the f-score for volumes 1 and higher remained very small.

Table 13: KNN Iterations Top Results

Clustering	Feature Set	K-nearest neighbors	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	Accuracy
SOM	3	5	0.80	0.20	0.06	0.07	66.7%
SOM	3	1	0.85	0.16	0.008	0.002	74.1%

vi. Support Vector Machine

The final machine learning algorithm implemented was a Support Vector Machine (SVM). Current literature contests that SVM is one of the most state-of-the-art supervised learning methods in terms of the prediction accuracy and its capacity to handle noisy data. The reason is due to the way classification boundaries are determined using this method – rather than developing a classifier which successfully finds some decision boundary within feature space, a separating hyperplane is computed which maximizes the margin of separation between the decision boundaries, accounting for noise that may not exist in the training data.

The lessons learned from the previous models were utilized to enhance the quality of the SVM model. The first model attempted to use only occurrences of call volumes given in the original dataset, which produced a constant prediction of a '1'. The reason was because the '1' volumes significantly outweighed the other classification values, and no training examples were provided for '0' volume. The next approach was to populate all the missing volumes with representative '0' values to introduce temporal features with a corresponding value of zero. This approach replaced the previous class imbalance from being dominated by single calls, to being dominated by zero call volumes. This introduced the need for an approach to balance the classes.

The use of duplicate samples added a weight component to higher call volumes, where the number of samples for a particular hour in a cluster was equal to the volume of calls during that hour. The number of samples for each of the other volume classes was then scaled to a reduced proportion. Many combinations were tested for feature selection, which included the values shown in Table 14. Of these features, the combination of day, hour and cluster code produced the best overall results.

Additional subsampling was performed to model and test using only subsets of the data characteristics. For example, only data from a single month was used to train a classifier and was then tested using only the test data of the same month. This approach was tested through subsampling months, clusters, and weeks. The best results were obtained using dataset 6, which consisted of only samples from the year 2003 without duplicates, and was randomly sub-sampled

to include 20,000 volume zero samples, 15,000 volume 1 samples, and 10,000 samples from volume 2 or higher.

Table 14: SVM Feature Selections

Set	Month	Hour	Day	Day of Week	Season	Week Number	Year	Hour of Year	Cluster Code
1(Base)	✓	✓	✓	✓	✓	✓	✓		✓
2	✓	✓	✓	✓					✓
3	✓	✓	✓						✓
4		✓	✓					✓	✓
5		✓	✓						✓

Table 15: SVM Dataset Parameters

Data Set	Year Range	Duplicates	Month Range	Cluster Range	Volume zero samples	Volume 1 samples	Volume 2+ Samples
1	2000-2003	no	All	All	10,000	10,000	10,000
2	2002-2003	no	All	All	15,000	10,000	5,000
3	2002-2003	yes	All	All	30,000	8,000	2,000
4	2003	no	1	All	248,668	55,198	11494
5	2003	yes	1	All	248,668	55,198	25,132
6	2003	no	All	All	20,000	15,000	10,000

The SVM model was tuned according to several key parameters. A Gaussian kernel was applied at various scales, which adjusts the window size of the Gaussian function. A scale of approximately 2.2 generally produced the best results. An option can be selected to standardize the data, which produced less accurate results in all cases. The box constraint level adjusts the "soft margin" of the classifier. Through investigation of the feature selections, it was determined that the data was not linearly separable, which necessitated the use of a soft margin for the fraction of misclassifications allowed between decision boundaries. This parameter produced the best results at a value between 5 and 10. K-fold cross validation was tested using 5, 10, and 15 folds. It was observed that increasing the number of folds beyond 5 did not affect the training results, possibly due to the model requiring only a small number of folds to pass the cross-validation test. For larger training datasets, a 20% holdout validation was used which produced

more accurate models and decreased the overall training time. The results of the best performing models are shown below in terms of individual F-scores and accuracy.

Table 16: SVM Iterations Top Results

Data Set	Feature Set	F-score "label 0"	F-score "label 1"	F-score "label-2"	F-score "label-3"	Accuracy
3	3	0.83	0.21	0.006	0.12	72.0%
3	4	0.86	0.13	0.02	0.04	75.14%
6	5	0.73	0.26	0.10	0.17	58.2%

The model trained using dataset 6 and feature set 5 was considered to produce the best results for the SVM models and compared to the other classification algorithm models. Although the accuracy overall was only 58.2%, the distribution of F-score values for each volume class was better than any of the other test results. The training accuracy of this model was 44.7%, with the training confusion matrix shown below. The model was trained using 45,000 observations and required 439.75 sec using a laptop with 8GB of RAM. The prediction speed was approximately 450 observations per second, which translated to around 12 minutes to fully predict the test dataset.

Figure 6: SVM Training Confusion Matrix

Overall, 183,964 out of 316,224 volumes were correctly predicted. Of the misclassified values, 39,639 were underestimated and 92,621 were overestimated. These results are also preferable as an overestimate is safer than an underestimate for demand planning purposes.

VI. Conclusion & Discussion

After implementing a collection of different supervised learning models to predict EMS call volumes, we found that our Support Vector Machine implementation performed the best with an overall higher accuracy and superior individual class accuracy distribution. More specifically, the top performing SVM yielded an overall accuracy of 57% and an F-Score of 72%. While other models came close, or surpassed, many of these metrics at an individual level none performed better overall than SVM. One of the most interesting conclusions we had with SVM was that using data for a single month, versus multiple months, produced a more generalized model for all months. Additionally, while inserting zero volume call records contributed more to our problem of class imbalance we found that it is necessary to provide an accurate representation of the data. For instance, running our SVMs without zero volume call records resulted in models that always predicted '1' for all sets of input features. One limitation of SVM we encountered was the runtime required to process larger datasets. Ideally, additional time and resources would have contributed to richer results and the ability to test more scenarios and strategies. Suggestions for future research include, working with larger datasets representing more recent time periods, incorporating additional features such as call disposition and demographics, as well as applying various under and over sampling techniques to correct the inherit class imbalance.

VII. Appendix

DATASET 1 – Missing 0 volume calls
Data Files: EMS_CALLS_SOM36_TRAIN_COMBINED.csv,
EMS_CALLS_KMEANS36_TRAIN_COMBINED.csv
Filter: Years = [2002, 2003]
Number of Observations: 159095

Data Sample

Month	Day	Day of Week	Year	Hour	Week Number	Season	SOM36_Clusters	Volume
9	3	3	1997	14	36	3	21	1
9	3	3	1997	14	36	3	34	1
9	3	3	1997	14	36	3	9	1
9	3	3	1997	14	36	3	15	1
9	3	3	1997	15	36	3	3	1
9	3	3	1997	15	36	3	21	1

Feature Exploration

Feature Name	**Min Value**	**Max Value**	**Mean**	**Standard Deviation**
Month	1	12	6.61	3.45
Day	1	31	15.66	8.79
Day of Week	1	7	3.93	1.96
Hour	0	23	12.95	6.11
Week Number	1	53	26.81	15.06
Year	2002	2003	-	-
Season	1	4	2.52	1.11
Cluster Number	1	36	18.47	9.81

Distribution of classes (Training)

Output Class (Volume)	**Frequency**
1	108532
2	36782
3	9807
4	3974

DATASET 2 – Considering 0 volume calls
Data File: EMS_CALLS_SOM36_TRAIN_ZEROS_COMBINED_FINAL.csv
EMS_CALLS_KMEANS36_TRAIN_ZEROS_COMBINED_FINAL.csv

Filter: Years = [2002, 2003]
Number of Observations: 654926

Data sample

Month	Day	Day of Week	Year	Hour	Week Number	Season	SOM36_Clusters	NewVolume
9	3	3	1997	14	36	3	1	0
9	3	3	1997	14	36	3	2	0
9	3	3	1997	14	36	3	3	0
9	3	3	1997	14	36	3	4	0
9	3	3	1997	14	36	3	5	0
9	3	3	1997	14	36	3	6	0
9	3	3	1997	14	36	3	7	0

Feature Exploration

Feature Name	**Min Value**	**Max Value**	**Mean**	**Standard Deviation**
Month	1	12	6.53	3.45
Day	1	31	15.73	8.80
Day of Week	1	7	3.99	1.99
Hour	0	23	11.61	6.86
Week Number	1	52	26.48	15.05
Year	2002	2003	-	-
Season	1	4	2.49	1.11
Cluster Number	1	36	18.48	10.32

Distribution of classes (Training)

Output Class (Volume)	**Frequency**
0	495831
1	108532
2	36782
3	9807
4	3974

VIII. References

1. Aringhieri, R., et al., *Emergency Medical Services and beyond: Addressing new challenges through a wide literature review.* Computers & Operations Research, 2016.
2. Channouf, N., et al., *The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta.* Health Care Management Science, 2007. **10**(1): p. 25-45.
3. Rajagopalan, H.K., et al., *Ambulance Deployment and Shift Scheduling: An Integrated Approach.* Journal of Service Science and Management, 2011. **Vol.04No.01**: p. 13.
4. Setzler, H., C. Saydam, and S. Park, *EMS call volume predictions: A comparative study.* Computers & Operations Research, 2009. **36**(6): p. 1843-1851.
5. Chen, A.Y., et al., *Demand Forecast using Data Analytics for the Pre-allocation of Ambulances.* IEEE Journal of Biomedical and Health Informatics, 2016. **20**(4): p. 1178-1187.
6. McConnel, C.E. and R.W. Wilson, *The demand for prehospital emergency services in an aging society.* Social Science & Medicine, 1998. **46**(8): p. 1027-1031.
7. Aldrich, C.A., J.C. Hisserich, and L.B. Lave, *An analysis of the demand for emergency ambulance service in an urban area.* American journal of public health, 1971. **61**(6): p. 1156-1169.
8. Kvålseth, T.O. and J.M. Deems, *Statistical models of the demand for emergency medical services in an urban area.* American Journal of Public Health, 1979. **69**(3): p. 250-255.
9. Siler, K.F., *Predicting demand for publicly dispatched ambulances in a metropolitan area.* Health Services Research, 1975. **10**(3): p. 254-263.