Prediction of Ticks in the Midwest

Cassidy Kohls
clkohls@uwm.edu

Matthew Voss
vossmc@uwm.edu

1. Executive Summary

Blacklegged ticks are the main source of the Lyme disease bacterium, Borrelia burgdorferi, in the Midwest[2]. The main culprit behind the spread of Lyme disease is immature blacklegged ticks, known as Nymphs. These hosts spread the disease to humans through bites and are tough to spot because they are only two millimeters in length! With the difficulty in spotting these pests, it can be challenging to limit the spread of Lyme disease. The infectious disease is accompanied by various painful symptoms such as severe headaches, rashes, and facial palsy, among others.

According to the U.S. Center for Disease Control (CDC)[2], 30,000 cases of Lyme disease are reported every year. Yet, it is estimated that 476,000 people may contract the disease. Although not all cases are documented because either people do not know that they have the disease or because reporting practices vary by state. For this reason, researchers have been trying to track the spread of blacklegged ticks in hopes of curbing the threat Lyme disease poses to the public; however, their work is limited to Logistic Regression and Cox Regression[4]. As a result, our group was inspired to improve the work already accomplished by these researchers to hopefully improve prediction accuracy in identifying counties that may have ticks. It also should be mentioned that the CDC reports that removing a tick early drastically reduces the probability of obtaining Lyme disease. Therefore, we hope to tackle the spread of Lyme disease by identifying which counties in the Midwest are acutely vulnerable to ticks so that individuals and authorities can detect the disease earlier and have the ability to make better-informed decisions.

2. Data Description

Both the data sets that describe the change in Lyme disease rates are structured and dynamic. The Midwest Lyme Invasion data set[3] was collected over a period from 1962 to 2016. Additionally, the Center for Disease Control's[1] records the evolution of Lyme disease cases from 2000 to 2019.

2.1. Midwest Tick Data

The Midwest Lyme Invasion data set is provided by scientists affiliated with the University of Illinois at Urbana-Champaign, who used secondary data like historical information such as local health department reports to build their data set. Their data set provides useful information, including the date of when the blacklegged tick (*Ixodes scapularis*) was first reported along with what state the incident happened in. As well as what type of area the tick was contracted in, for instance, if a tick was picked up near the forest or a stream. Most importantly, the data reveals if Lyme disease cases are present in specific counties. The insight this data can provide assists with predicting outbreaks of blacklegged ticks so that hospitals can provide better care and authorities can make better-informed decisions.

2.2. CDC Tick Data

The U.S. Center for Disease Control documents the number of ticks over the years through secondary data from state and local health departments. Their data sets yield information regarding the number of reported cases within states and, more specifically, counties. This data set will be used to test the model; therefore, the insight it can provide is whether the model is accurate. Overall, this will help develop a better forecast and provide more value to the authorities.

3. Analysis Plan

There are multiple methods available to analyze the Midwest Lyme Invasion Data including logistic regression, the histogram gradient boost, random forest classifier, cluster analysis, and neural networks.

3.1. Logistic Regression

We planned on using this as our base method to classify if a county has blacklegged ticks present (1) or not (0) based on various independent variables, which include:

- The time when the tick was first observed
- The state it was observed in
- The percentage of forest coverage in the county
- The size of the rivers within the county
- If an adjacent county has been invaded

Once the model is built, we will apply grid search cross-validation to estimate the performance of the model. By using this method, we will be able to detect any problems or selection bias within our model. Essentially, this model should be able to predict the next counties within the Midwest that could become invaded with blacklegged ticks. However, one challenge that we could face is not knowing whether or not an adjacent county has been invaded because future data is not available.

3.2. Histogram Gradient Boosting Method

This technique improves prediction power by requiring a weak learning model to master its mistakes to become a strong learning model. After building a model on our training set, a second model will be built to fix the errors in the first model. This process continues until all the errors are reduced. It should be mentioned that one difficulty we could encounter with this method is that it is susceptible to meaningless data, which can lead to our data being over-fitted.

3.3. Random Forest Classifier

The random forest classifier is a simple supervised method for non-linear classification. Basically, it builds multiple decision trees (known as the forest) out of the data. Once that step is complete, each tree "votes", and they reach a collective agreement. Based on this decision, the model produces a final output. The more decision trees built within the forest, the more accurate the prediction becomes. At the same time, the more decision trees that are built, the more time it takes to train the model, which could become a challenge. Overall, this model could be effective for our project if the other models have poor prediction accuracy.

3.4. Cluster Analysis

Lastly, cluster analysis will assist us in finding patterns and classifying the data into similar groups, which will help identify if certain geographical areas are linked with high levels of blacklegged ticks. Nonetheless, one challenge we need to keep in mind is that the cluster sample may not be representative of the entire population, which could affect our ability to predict the spread of blacklegged ticks based on this analysis. Once we have determined the optimal number of clusters to use for the K means cluster analysis we will prepare a silhouette plot to understand the clusters.

3.5. Neural Networks

Neural Networks offer the ability to generally fit non-linear data with great precision. By expanding the number of hidden layers, types of layers, and the time to reduce the cost function it offers a promising ability to predict counties affected by ticks. The main promising feature of neural networks are their accuracy while their interpretability is often a point of concern especially for people in the social sciences. We intend to use LSTM layers to hopefully increase the bias for recent data and then flatten the dimension with a linear layer.

4. Expectations

We expect our analysis to provide an accurate prediction of migration patterns of the blacklegged ticks using our specified independent variables, invasion status. However, our model could have a poor prediction accuracy, or our independent variables could have no impact on invasion status. Therefore, our insight would not be useful for authorities that deal with outbreaks associated with the blacklegged ticks. If our prediction accuracy are not very accurate, we still anticipate that by building these models, entomologists will gain a better insight into the understanding of the importance of geographic features play on the migration of blacklegged ticks. Our models will allow entomologists to understand which Midwestern counties have a similar geographical build. This information can then guide in satisfying the growing human population well at the same time providing ecosystem stability. In the data set, features include physical characteristics of the county, providing more acute details. In addition, the data set includes the state as an indicator variable. While it is simple to find summary statistics by the feature level, our models will hopefully guide entomologist to make the best decisions for society regarding the presence of blacklegged ticks within certain areas. As of now, researchers can find a trend of ticks one at a time. However, when putting these features together the picture is more difficult to decipher. As a result, we hope that by using a variety of methods to analyze the Midwest Invasion Data, it will help make the information less difficult to decipher as well as assist researchers in understanding what variables have the most impact on how blacklegged ticks spread. Overall, we expect that our models to aid authorities by allowing them to make informed decisions.

5. Recommendations

Generally, deep learning models are excellent at finding precise predictions. Therefore, our group was enticed by this classification of model and agreed to experiment with these models. Quickly, the data set proved to be far too small to obtain an accurate prediction from the deep learning models. The LSTM layered network showed little hope but a few percentage points lower than the accuracy of logistic regression. Evidently, neural networks are not the correct path to take in this project.

With the constraint of a small data set, the next intuitive choice is a machine learning classifier for the project. These models are more robust with small data sets than deep learning models and are also slightly more interpretable. For instance, in the random forest classifier, it is much simpler to understand a decision tree versus viewing individual cells of a neural network. Unlike deep learning models, which are notoriously bad with interpretation, decision tree-based models are an improvement. While machine learning models lack the interpretability of a linear model, these models offer higher accuracy in predicting which counties may have ticks reported. Although interpretability is important in understanding tick migration, prediction accuracy offers benefits for entomologists who wish to understand the spread of ticks.

6. Implementation

We planned to try two of the three subsets of artificial intelligence, mostly to figure out which one improved our model's prediction accuracy and to gain experience in each subsection of the field. In unsupervised learning, we implemented a K-means cluster analysis. In supervised learning, we implemented logistic regression, random forest classifier, histogram gradient boost classifier, and a few neural networks. We also implemented a standard scalar to make the features more similar to each other.

6.1. K-means Clustering

Our group fitted a K means cluster on the feature set in hopes of finding a pattern with this method. First, we needed to discover the optimal number of clusters for the data. We looped through the K-means class from Sci-kit Learn to find a balance between inertia and the number of clusters. As seen in Figure 1, inertia is slowing after 2 clusters. This fits our assumptions of the data since we know the data is broken into whether the data has blacklegged ticks or none. Next, we wanted to see what the clusters looked like on the feature set. Unfortunately, we could not draw this graph with 15 dimensions. Thus, we drew a silhouette plot in order to understand the fit of the cluster of each point assigned to that cluster. We used the silhouette samples metric (also from Scikit learn) and ordered the silhouettes with a bar graph, as seen in Figure 2.

6.2. Logistic Regression

Logistic Regression is created with the following equation:
$\hat{\pi_i} = \frac{exp(\hat{\beta_0}+\hat{\beta_1}x_1+...+\hat{\beta_p}x_p)}{1+exp(\hat{\beta_0}+\hat{\beta_1}x_1+...+\hat{\beta_p}x_p)}$ This model only accepts binary responses. To find the accuracy of this model on our data we implemented grid search to find the training and validation accuracy. With 10 k-fold cross validation, we found the validation accuracy was 74.5 percent.

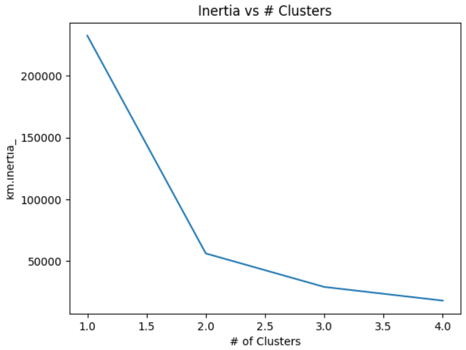


Figure 1. Finding number of clusters with inertia

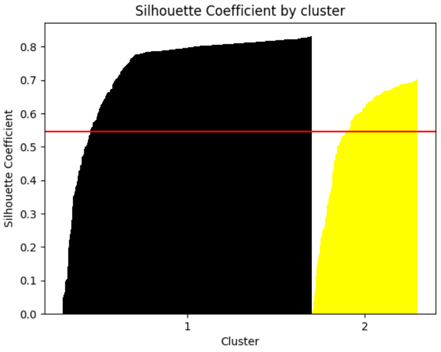


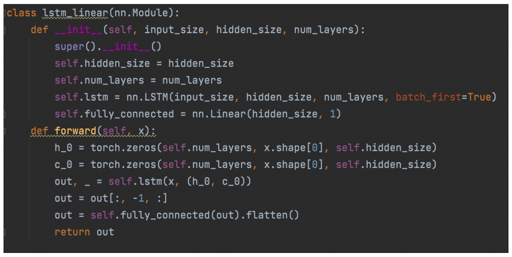
Figure 2. Understanding the distance of points from the center of their respective cluster

6.3. Random Forest Classifier

Random forest is a special form of a decision tree that uses bootstrap on both features and data. Generally, this method produces great results, and we found a validation accuracy of 73.9 percent. We used grid search with 10 k-fold cross-validation to obtain this accuracy. The parameters to obtain this accuracy are entropy for the criterion, max depth set to 8, min samples leaf to 2, and n estimators to 79.

6.4. Histogram Gradient Boost Classifier

Histogram gradient boost classifier is another form of a decision tree that sequentially adds branches with some loss function. This method performed better than the AdaBoost model, so we decided to use this method for a gradient boost



```
class lstm_linear(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers):
        super().__init__()
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True)
        self.fully_connected = nn.Linear(hidden_size, 1)
    def forward(self, x):
        h_0 = torch.zeros(self.num_layers, x.shape[0], self.hidden_size)
        c_0 = torch.zeros(self.num_layers, x.shape[0], self.hidden_size)
        out, _ = self.lstm(x, (h_0, c_0))
        out = out[:, -1, :]
        out = self.fully_connected(out).flatten()
        return out
```

Figure 3. LSTM layered neural network code

classifier. With an accuracy of 73.94 percent on the validation set, it performed a little poorer than the random forest model. This accuracy was also found using grid search cross-validation with 10 k-folds. Overall, the best parameters were 215 max bins, 5 max-leaf nodes, and 10 min samples leaf.

6.5. Long Short-Term Memory Layered Neural Network

The last model our group tried was an LSTM-layered neural network. We built the model using the Pytorch library[5]. We built the neural network as a feed-forward model, as seen in Figure 3. With poor performance we quickly abandoned this model for traditional machine learning models due to the size of our data.

7. Results

With poor performance while training a few machine learning models, we decided to use the random forest classifier for our model. Unfortunately, the random forest classifier had just 74 percent test accuracy. Overall, our prediction accuracy fell short of the original study. The researcher's model resulted in a 90+ percent accuracy, while our model was fitting poorly on multiple methods. In the end, we used three modules: Pandas, Sci-kit Learn, and Pytorch, to produce our results.

Ultimately, due to the poor prediction accuracy of our models, we could not aid individuals or authorities in detecting the disease early or give them the ability to make better-informed decisions. However, by implementing the options discussed in the next steps, our model's accuracy may improve and could potentially end up helping control and prevent the infectious disease.

8. Next Steps

Since one of the group members uses Pytorch professionally, we thought we would be able to reach a high prediction accuracy. However, there was an issue with the data loader. Therefore, an additional step that could be implemented is combing through the LymeDisease Class to try and find the bug that led us so far astray. Another possible step is implementing the tensor flow module, which has similar neural networks with a slightly different configuration. It also should be noted that there seemed to be an issue scaling the data with the sci-kit learn standard scalar, so trying to fix the scalar may have produced better results. Besides fixing issues within Pytorch, additional data on variables such as temperature or tick carrier migration within deer and other animals, as well as data provided on all counties for each time period, would have been beneficial to our model. Ultimately, if we had more than approximately 1,000 lines of data related to the invasion of blacklegged ticks, it would have drastically improved prediction accuracy.

We also recommend using R for additional analysis because unlike Python, which is dynamic, R is useful for determining small data sets. For instance, DPLYR is an incredible tool to manipulate data quickly and easily, which is sorely missing in Python. Another capability of R that can be seen as a strength is the ease of use in packages such as MASS at such a high level. Meaning R can produce a SoftMax regression much quicker and in fewer lines of code. On the other hand, Python offers a more developed suite of libraries to use for machine learning. For example, Scikit Learn, Pytorch, Tensorflow, and Keras libraries are capable of building deep learning or machine learning models. While we implemented a few classifiers from the Scikit Learn library, such as “GridSearchCV()”, random forest, and histogram gradient boost, there are many more classifiers that could be used to better obtain the model. However, due to the dataset’s small size, it is unlikely deep learning models will have enough data to create robust models. Therefore, traditional machine learning algorithms are ideal for this scenario.

References

[1] CDC. County-level lyme disease data from 2000-2019.

[2] CDC. Lyme disease.

[3] A. Gardner and B. Allan. Gardner midwest lyme invasion data 2020-11-18.

[4] A. Gardner and e. a. Pawlikowski, N. Landscape features predict the current and forecast the future geographic spread of lyme disease.

[5] PyTorch. Long short-term memory.