SAND2021-4910R

TAFI/Kebab End of Project Report

Kyra Wisniewski, Katrina Ward, Kanad Khanna, Mark Rintoul
Sandia National Laboratories

March 2021

Abstract

This report focuses on the two primary goals set forth in Sandia's TAFI effort, referred to here under the name Kebab. The first goal is to overlay a trajectory onto a large database of historical trajectories, all with very different sampling rates than the original track. We demonstrate a fast method to accomplish this, even for databases that hold over a million tracks. The second goal is to then demonstrate that these matched historical trajectories can be used to make predictions about unknown qualities associated with the original trajectory. As part of this work, we also examine the problem of defining the qualities of a trajectory in a reproducible way.

1 Introduction of Problem Space

1.1 Prediction

The previous work on the problem of trajectory prediction can be divided into two categories based on approach taken: Markov models and neural networks. Yet, these solutions are unsatisfactory for a number of reasons. Markov model-related solutions work best when the prediction space is discrete (prediction of roads for driving data). However, this discrete approach doesn't apply well to less constrained domains (such as maritime/boat trajectories) where an extremely vast search space would need to be discretized [1–3]. On the other hand, deep learning approaches (the most successful being Long Short-Term Memory (LSTM) neural networks, because of their ability to handle time series data) produce more accurate predictions than Markov modeling methods. Deep learning has been used for both next step prediction [4,5] and destination prediction [6–10]. However, like Markov models, most deep learning approaches in this domain also resort to discretizing the geographical landscape to deal with the domain's sparsity issue. For example, the majority of physical locations (e.g., the middle of a lake or off-road areas for terrestrial vehicles, or non-airport locations for aircraft) are unlikely destinations, and yet the total number of possible destinations is still very large. In addition, deep learning approaches tend to

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

lack explainability, of which is required for wide adoption and real-world use by analysts. It is worth noting that there are also some attempts to apply similarity measures to the problem of prediction [11]. Also, it is possible to simply use a polynomial extrapolation or other mathematical techniques that don't take advantage of historical information to predict the future location of the trajectory. But these techniques don't capture the complex human decisions that are often a factor in trajectory behavior.

2 Statement of Work

2.1 Alignment

Can a trajectory be aligned to other trajectories that are similar in shape and relative location? For example, given an observed trajectory can we identify historical trajectories that have traveled along a path, a portion of which is similar to the path of this observed trajectory?

2.2 Prediction

If an object's motion is captured only for a short period of time, can we use historical information to fill in the unknown qualities of the past part of the trajectory and predict its future qualities? For example, given an observed trajectory one possible goal could be to correctly predict its origin and destination locations. Alternatively, another relevant problem is correctly predicting where an observed trajectory will be in a specified amount of time, such as predicting where an aircraft will be 15 minutes in the future. We focus on these problems by leveraging the alignment algorithm and by using historical data to inform predictions. Namely, we ask which historical trajectories have a path similar to the given observed trajectory. Note that there are two different scenarios that can result:

1. There is at least one historical trajectory that is suitably aligned to the given observed trajectory. In this case, it is possible to develop weights based on the alignments such that the results can be ranked before they are used.

2. There are no historical trajectories that are similar to the given observed trajectory. This is an important null result as it indicates the current trajectory is following a path that does not have a historical match.

Figure 1 shows an example of the historical trajectories that align with the short (red) observed trajectory over Pennsylvania.

The output of an origin/destination trajectory prediction algorithm can take multiple different forms. For example, as a single origin/destination pair with the highest likelihood of correctness. Alternatively, a ranked list can be returned like a typical internet search for evaluation by an expert analyst.

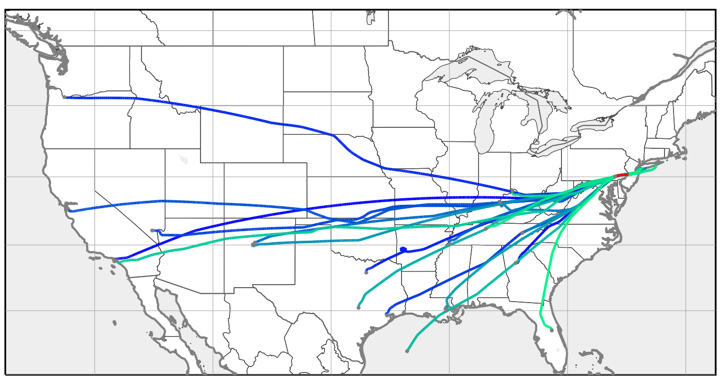


Figure 1: This example represents the prediction of the origin/destination of the red observed trajectory and a visualization of the 19 suggestions for this problem based on our prediction algorithm. We correctly predict the flight is going from JFK - LAX using 5 days of historical data, meaning that the origin destination pair JFK-LAX was given the highest weight and appears first on the list of possible origin destinations. The darker the trajectory the more weight the origin destination pair receives from the algorithm.

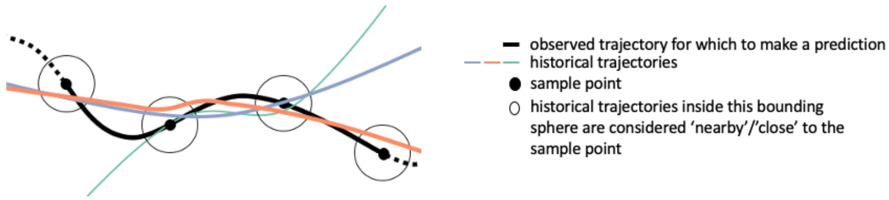


Figure 2: The orange historical trajectory is close to each of the observed trajectory sample points, and so is considered well-aligned with the observed trajectory.

3 Milestones and Results

3.1 Alignment

In order to find historical trajectories that are well-aligned with an observed trajectory, we developed an algorithm that works as follows. We represent the observed trajectory as a list of k evenly spaced points in space. We refer to these points as sample points. For each of those sample points, we search the historical data to find points of historical trajectories that are nearby as defined by the point-trajectory distance. A historical trajectory that is within a specified distance, d, of all k sample points of the observed trajectory, is considered to be well-aligned with the observed trajectory (see Figure 2). It is worth noting that we measure the distance between each point of the observed trajectory to the segments of each historical trajectory (not points on the historical trajectory). This helps to control the effects of sampling rate.

To make querying this historical geospatial data efficient, we use a spatial index for fast spatial searching. The spatial index allows for efficiently find what points are near (within some bounding box around) a given reference point. For this work the R-Tree spatial index is used [12]. Leveraging this structure allows us to store every point of every historical trajectory and maintain a look up time complexity of $O(\ln(n))$, where n is the total number of points in all of the historical tracks in the spatial index.

3.2 Prediction

We developed an algorithm that makes predictions for a given observed trajectory by finding historical trajectories such that part of the historical trajectories follow a similar path to that of the observed trajectory. To do this, we first leverage the alignment algorithm to find the historical trajectories that are well-aligned with the observed trajectories. Then, the prediction algorithm accounts for direction in the following manner: If a historical trajectory is well-aligned with the observed trajectory, we check to see if the historical trajectory and the observed trajectory are going in the same direction. Figure 3 describes this process. Once similar historical trajectories are found through the alignment process, those flights can be used to potentially predict qualities that the observed trajectory has, such as specific object type, mission type, etc. We focus on origin/destination prediction in this work as a compelling example of one type of prediction that has a strong geospatial driver.

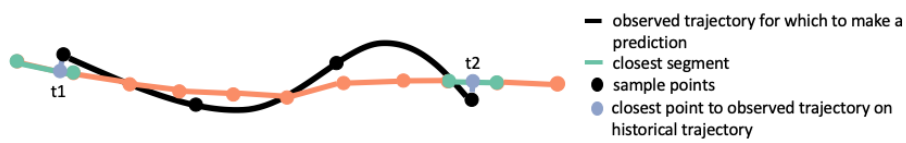


Figure 3: To determine if an observed trajectory (black) and a historical trajectory (orange) are going in the same direction, do the following. Represent the historical trajectory using segments. Find the closest segment of the historical trajectory to the first and last points in time of the observed trajectory. Interpolate along these two segments to find the closest point in space on the historical trajectory to the first and last point of the observed trajectory. Let t_1 be the time of the point on the historical trajectory closest to the first point of the observed trajectory in space. Let t_2 be the time of the point on the historical trajectory closest to the last point of the observed trajectory in space. If $t_2 - t_1 > 0$ then the historical trajectory and the observed trajectory are going in the same direction.

As shown in Figure 4, checking for consistent direction does not significantly impact the number of correct origin/destination pairs predicted, but it removes extraneous predictions resulting in the correct origin/destination being predicted with fewer guesses and with higher confidence. If a historical trajectory passes both the alignment and direction requirements, its origin/destination pair is returned as a possible prediction for the observed trajectory's origin/destination. Note that particularly pathological trajectories could potentially visit the sample points in an odd order. We did not find any examples of this in our data set, and it would not be hard to filter for this by looking at the direction between each pair of neighboring points sequentially.

Given a list of origin/destination predictions, it is important to consider how to order the list to best provide information on the likelihood of each origin/destination pair being the correct origin/destination of the observed trajectory. We assign each origin/destination pair a weight, which is the number of trajectories with the given origin/destination pair normalized by the number of total trajectories that are close to all k sample points of the observed trajectory. We experimented with taking into account the distance of each historical trajectory with a given origin/destination from the observed trajectory in the weight of the origin/destination pair. Most of the weighting schemes gave relatively similar answers, assuming they were a monotonically decreasing function of the distances between the observed trajectory sample points and the historical trajectories. Simple functions such as equal weights or linearly decreasing weights worked as well as more complex weighting schemes.

We ran experiments on the origin/destination prediction algorithm to determine how different factors influence its performance. Figures 4, 5, 6, and 7 detail these results. In the experiments, the weight of an origin/destination pair is the integrated weight (sum of the weights) of those before it in the prediction list. Integrated weight is used because the predicted integrated weight of an origin/destination pair is a good analog to predicted position of an origin/destination pair on the predictions list. Additionally, the correct origin/destination pair's position on the prediction list can be thought of as the number of guesses needed to get the correct origin/destination pair.

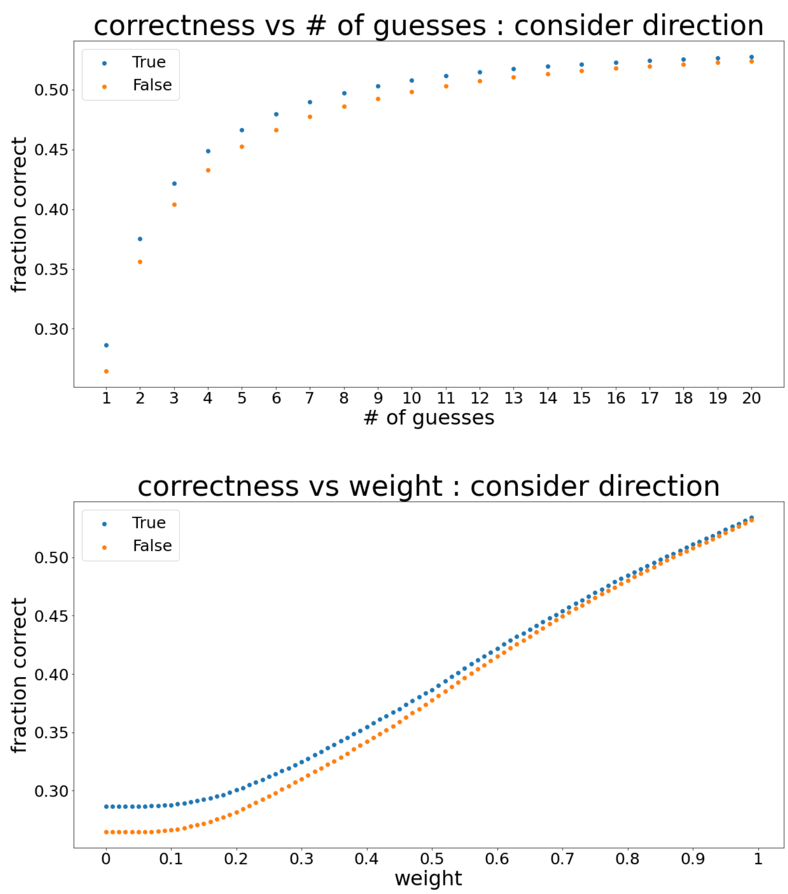
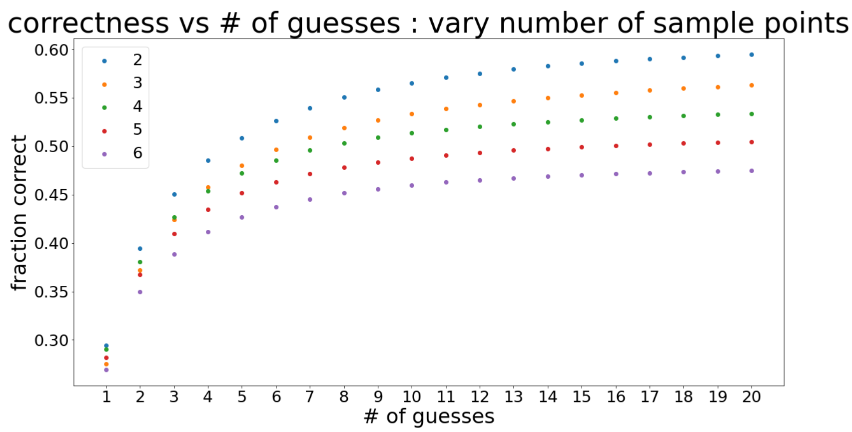


Figure 4: Experiments run on 5 days of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. In this case whether or not the direction check was required is varied.

Figure 4 shows the benefits of checking that a historical trajectory and the observed trajectory are traveling in the same direction. When travel direction is considered, the position of the correct origin/destination appears closer to the top of the prediction list and with less integrated weight. Both improvements get smaller as the position on the list and weight increase.

Figures 5 and 6 show how the prediction algorithm performs when the desired matching precision is varied (the greater the number of sample points and smaller the neighborhood distance, the more precise the matching will be). This is the classic trade-off of setting the threshold for false positives higher in order to get more true positives. When the number of sample points increases we see observable improvement on the correct origin/destination pair's location on the prediction list and its integrated weight. Varying neighborhood distance explores the predictive power of the algorithm as a function of how close we require the historical matches to be to the observed trajectory. Intuitively, one would expect that as you require the distances to be smaller, you will have a higher ratio of true positives to false positives and your predictive power will be greater. However, the graph shows the opposite for most of the distance choices used in the experiment. The subtle reason for this is related to the data set that we used for training and testing. In this data set, at the 10km neighbor distance, we have already captured most of the correct historical matches and removed the incorrect ones. If we decrease the neighbor distance from 10km, we remove more correct answers from the results compared to incorrect ones. However, at the 25km neighbor distance, we do see the expected trend. For less guesses and weight, the 25km neighbor distance captures fewer of the correct predictions compared to the other neighbor distances, but as we move farther down the prediction list we reach a point where the 25km neighbor distance captures more of the correct predictions compared to the other neighbor distance (the point at which 25km line intersect the others in both graphs). This indicates that care should be taken in choosing this adjustable parameter for prediction purposes. Automation of this choice is something that we would like to investigate in the future.

Figure 7 shows how the prediction algorithm performs relative to the length of the observed trajectory as compared to its complete length. When the subtrajectory in the experiment is constructed from a larger percent of the historical trajectory it is equivalent to having more data for the observed trajectory in a real-life application of this algorithm. When we have more data for the observed trajectory, we see observable improvement in the correct origin/destination pair's location on the prediction list and its integrated weight, but only to a point. If too much of the observed trajectory is used for the prediction it may not match with any of the historical trajectories due to the increasingly stringent conditions of matching over longer distances, even with historical trajectories that have the correct origin/destination. The data shows that you are much more likely to *eventually* find the correct match from a smaller fragment since that will return many more results, but you will also have more false positives. As the fragment size increases, you eliminate more of the false positives and better predictions can be made. However, if the fragments become



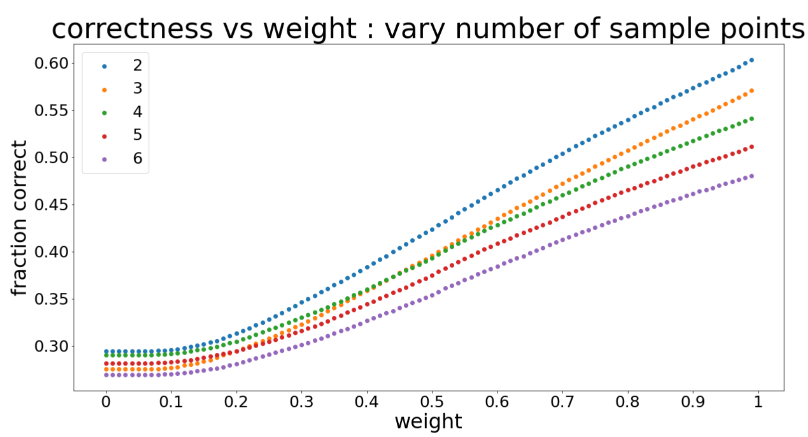
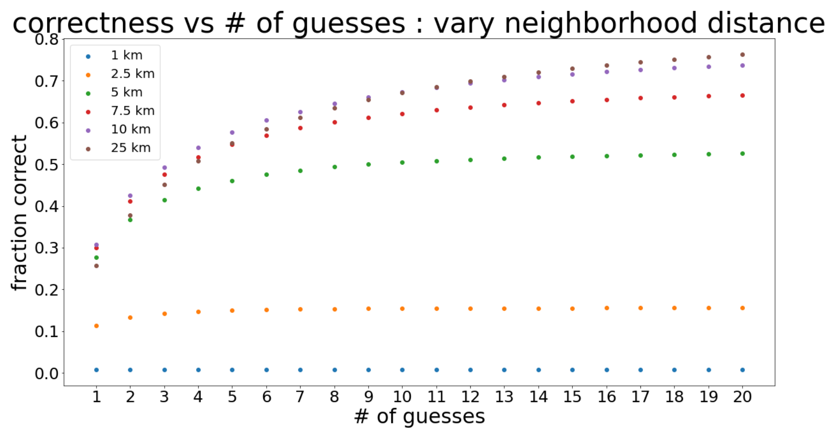


Figure 5: Experiments run on 5 days of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. In this case the number of sampled points used to represent the sub-trajectory was varied.



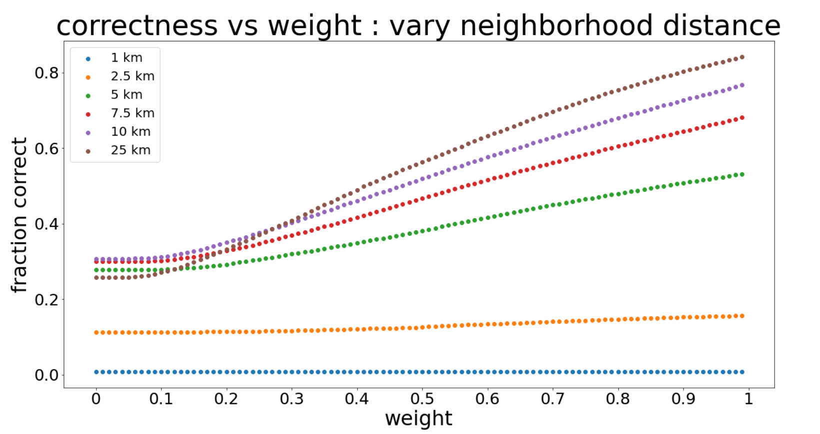


Figure 6: Experiments run on 5 days of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. In this case the size of the neighborhood distance used to find nearby historical trajectories.

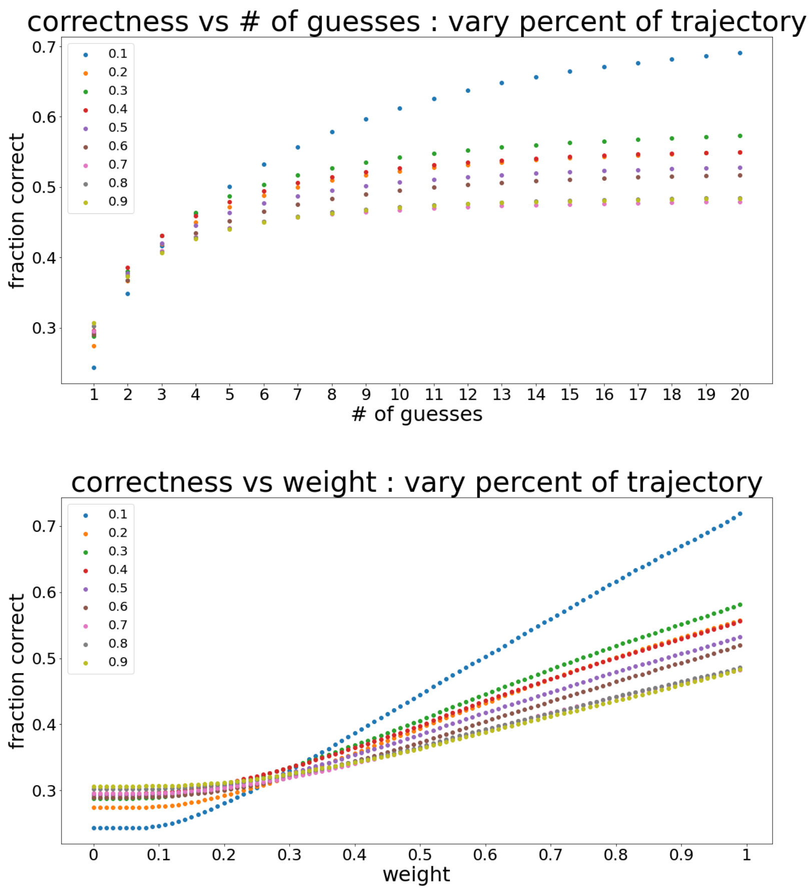


Figure 7: Experiments run on 5 days of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. In this case the percent of the trajectory used as the sub-trajectory was varied.

too large, you begin to then match fewer true positives and the predictive power decreases.

	1 Day	5 Days	1 Month
Number of trajectories	53,700	225,382	803,298
Percent Correct	.31	.57	.88

Table 1: Experiments run on different amounts of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. Experiments were run by representing a 100km sub trajectory with 4 sample points.

	1 Day	5 Days	1 Month
Number of trajectories	53,700	225,382	803,298
Time (s) to predict origin/destination	0.114	0.443	2.979
Time (s) to predict location	0.146	0.518	4.387

Table 2: Experiments run on different amounts of historical data from flights in US airspace. 1000 random trajectories were chosen from the 1-day data set and a 100km sub-trajectory represented by 4 sample points was randomly chosen from each. An origin/destination prediction and a location prediction were made for each of these sub-trajectories. The average time to do one prediction is recorded above.

Providing information about weights along with the list of possible predictions is valuable for an analyst. Without the assistance of the weights an analyst may constrain their analysis to some number of places on the prediction list, a natural example being 5 origin/destination pairs. However, consider the possibility where only the first origin/destination pair has a significant weight. The analyst can streamline their search. Alternatively, consider the possibility where the top 6 origin/destination pairs all have similar weights. The analyst should not use their standard place cut off heuristic (top 5) in this case.

The power of the alignment algorithm enables the prediction algorithm to be used to predict where an observed trajectory will be in specified amount of time in the future. Consider a historical trajectory that passes the alignment and direction requirements of the prediction algorithm for an observed trajectory. We find the closest point on the historical trajectory to the front (most recent) point of the historical trajectory and then move forward in time x minutes from this point on the historical trajectory for a prediction of where the observed trajectory will be in x minutes (Figure 8). Weights are then assigned by trajectory, instead of origin destination pair. Figure 9 shows an example of future position prediction.

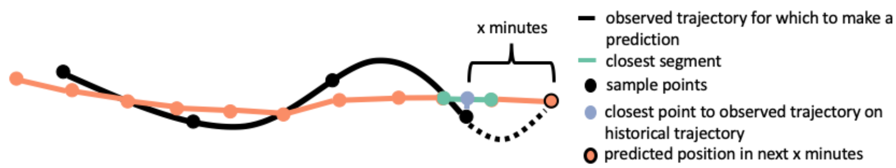


Figure 8: Prediction Approach. We first represent the orange historical trajectory using segments, and then find the closest segment of the historical trajectory to the last point of the observed trajectory. We interpolate along this segment to find the closest point on the historical trajectory to the last point on the observed trajectory. Finally, we traverse x minutes on the historical trajectory for a prediction of where the observed trajectory will be in x minutes.

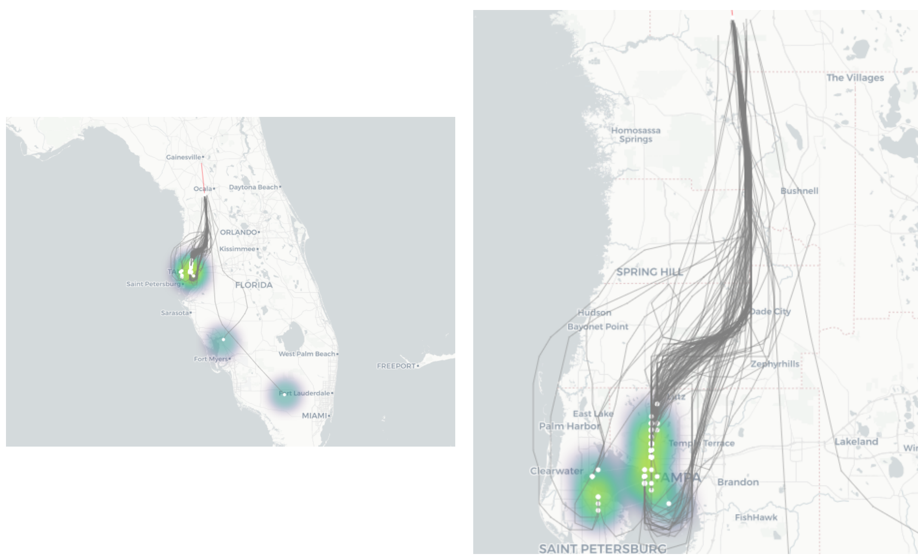


Figure 9: Example prediction of the position of the red observed trajectory in 30 minutes. The visualization of the suggestions for this problem are based on our prediction algorithm. The colors of the heat map correspond the weights of the predicted point. Green areas have more points with higher weights, purple areas have less points with lower weights. Historical trajectories that adequately match the observed trajectory appear in grey.

4 Additional Progress and Results

There were a number of research topics that emerged either as areas that were direct extensions of the statement of work or as later requests from the sponsor. We describe those efforts in this section. In general, these were smaller, exploratory efforts that we only briefly explored. A deeper exploration is left as potential next-generation work.

4.1 Motion Event Definitions

While communicating about trajectory shapes and motion events, it has become apparent that how we describe and/or define those motion events can be different from how others do the same. The differences can be significant enough to cause miscommunication and potential errors between different entities in the track community. Therefore, we are working towards a standard set of definitions that we, as a community, can agree upon to make communicating about motion events easier, faster, and more reliable.

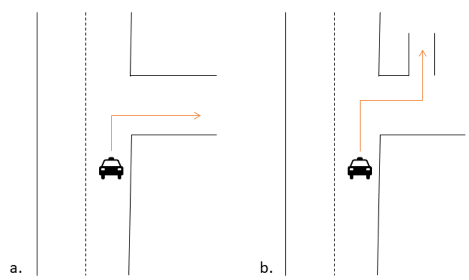


Figure 10: Some motion events are simple to define such as the vehicle in a making a simple right turn. Others are harder to define such as the vehicle in b. The problem becomes more complicated with different types of vehicles.

Let us first define what we mean by a motion event. When we observe an object in motion, we can detect events such as a start, stop, and/or turn. These types of events are motion events. They can be fairly simple or complex as seen in Fig. 10. The problem becomes more complex when we consider different types of vehicles and terrains. We determined that the most basic definitions would be: start, stop, pause, straight, left turn, right turn, u-turn, and possibly acceleration and deceleration. Any other events could be described as a combination of these events.

When beginning the investigation into motion events, we reached out to others in the track community. We received a strong response from BAE Systems, Kitware, MIT, Mitre, and other groups within Sandia. What we found is most researchers create their own definitions for their own systems, even within the same institution. Though almost all define their motion events in terms of speed, time, and radius of the event. BAE Systems had the most thorough set of definitions what would serve as a strong starting point. However, they have requested their information remain proprietary unless settled as the standard. They remain very willing to engage in the discussion and decisions.

The naive solution is to define a set of definitions that are universal to all vehicles. While this was the primary goal when the problem was presented, it quickly became obvious this was not feasible. If we consider something simple such as a left turn and we consider all vehicles together, then we must create a definition that encompasses the turn of a small motorcycle and a large cargo plane, while also being able to distinguish it from other motion events of either. The definition becomes too loose and multiple events have overlapping definitions.

The next logical consideration is to separate vehicles into types: ground, air, water. However, we again have the issue where even within the same category there are vehicles with very different profiles and motion capabilities. For example, if we consider marine traffic, a cargo ship moves very different from a small fishing boat.

Our final assessment is to group vehicles by class, where each class has a similar type of motion constraints. For example, we could have the following groupings: large ships (cargo, military, etc.), large planes (cargo, intercontinental, etc.), small ships (fishing, leisure, military), motor vehicles, medium or small planes.

We also looked at the difficulty in convincing an entire community to adopt a set of definitions where they had already decided on their own in their systems and where they had little say in the definitions we developed. In the last Trajectory Community of Practice meeting, we spoke about the work on motion event definitions and received defensive feedback with some suggesting that we challenged their own work. Many were interested in being involved in the definitions decisions, however. With this information, we have developed an overall process that we believe would create a set of standard definitions for motion events while allowing the community to be a part of the process and more likely to adopt the final result:

- Create a comprehensive list of types of vehicles the community is interested in
- Engage the community to use their knowledge to categorize the vehicles into similar groups based on their experience
- Develop a set of definitions for each category based on current definitions and understandings

With community members involved in some of the decisions, and their own definitions and work being a part of the final product, we believe we can develop a set of motion event definitions that are comprehensive and the rest of the track community can feel they have some ownership of and be more willing to adopt.

4.2 Prediction and Weighting

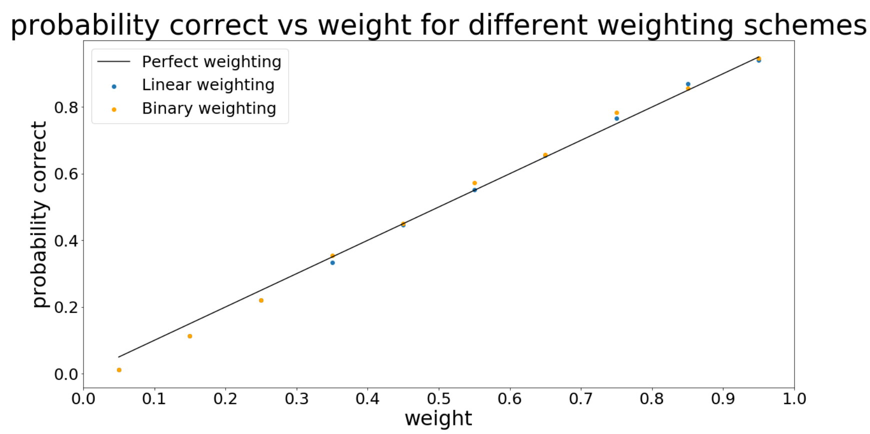


Figure 11: This experiment was on one month of historical data from flights in US airspace. For every trajectory in the data set, an origin/destination prediction was made for a randomly selected sub-trajectory of that trajectory. Two weighting schemes are considered and compared to a perfectly correlated relationship between weight and probability. Probability correctness is derived here from the number of times a predicted origin/destination pair with a particular integrated weight was the correct answer.

For historical training sets that are large enough to have a representative sample of the trajectories of interest, the relative fractions of the well-aligned trajectories can be used to make a rough prediction about the accuracy of the guesses using the weights of the matched trajectories. We tested this hypothesis by evaluating two different trajectory alignment weighting schemes and, using those weights, derive the probability that the guess would be correct. This was done by using one month of training data as the historical data set, and calculating weights based on two different schemes. The first scheme was to calculate a weight for each match using a linear function, going from 1.0 to 0.0 based on how far away the farthest sample point of the sample trajectory was from the historical trajectory of interest. The value was 1.0 if the furthest sample point was coincident with the historical trajectory and decreased linearly to 0.0 at the end of the range of interest (5km in the example shown). The second scheme was to use a binary weight, where a trajectory counted as 1.0 if the furthest sample point was in the range of interest, and 0.0 otherwise. The weights for all of the matches were then summed and this was used to normalize the sum of the weights to 1.0. The weights for each origin/destination pair were also summed, giving a total weight for each origin/destination pair. Note that we only used samples that had at least 5 well-aligned trajectories in the historical database so that there were enough representative samples.

We then binned, as a function of weights in bin widths of 0.1, the fraction of matches that were correct for each weight. The goal was to have the weights correspond to a probability that the guess was correct. We consider this a 'perfect' weighting. Figure 11 shows the results of our experiment. The predicted probability that was based on the weights turned out to be surprisingly accurate. This effect was almost entirely captured by the binary weighting scheme, indicating that more sophisticated measures that took into account how far away a sample trajectory was from the historical trajectories was probably not necessary when there is enough historical data. This notion of being about to capture a probability that a guess is correct is very important to real-life scenarios where confidence is key.

4.3 Deep Learning

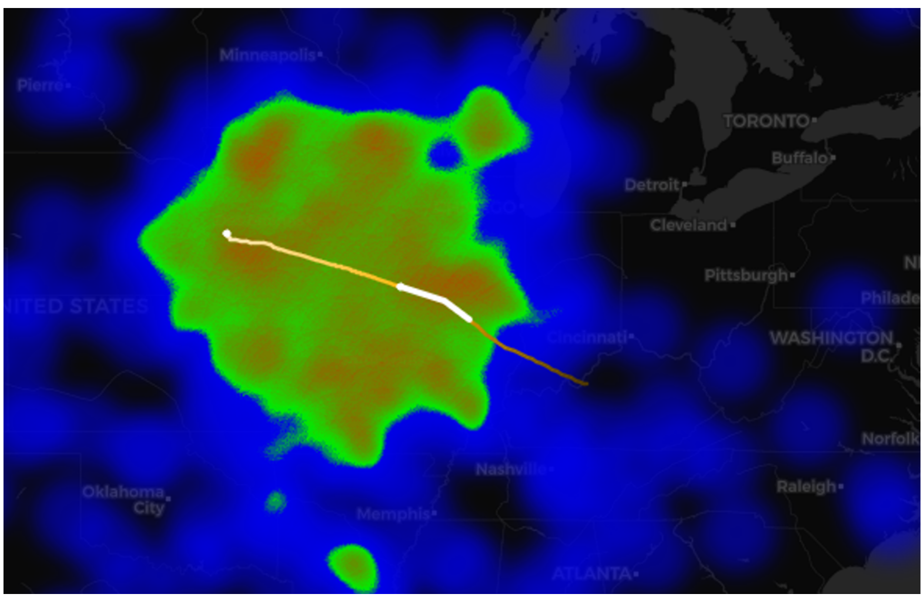


Figure 12: The predicted destination of a trajectory, calculated by a simple neural network model. The model only sees the segment highlighted in white and does not know what part of the full trajectory it is observing. The likelihood of a location being the destination is shown as red (high), green (medium), blue (low), or black (very low).

We also explored the implementation of a deep learning model capable of producing nuanced, yet simple and visualizable predictions. We designed a mixture density network [13] to read GPS data from a trajectory and output a multimodal probability distribution over the geographical landscape, representing the model's hypothesis of where the destination (or origin) might be. The model was trained to output a von Mises-Fisher mixture distribution, analogous to a Gaussian mixture model over the surface of the Earth. A mixture model is the linear combination of some number of individual sub-distributions, such that each sub-distribution belongs to some (frequently the same) parametric family. Thus, whereas a standard neural network outputs the most likely conditional mean

$$\arg\max_{\bar{\boldsymbol{y}}} p(\bar{\boldsymbol{y}}|\boldsymbol{x})$$

of the target variable, $\boldsymbol{y}$, a mixture density network produces a mixture distribution over the target variable; i.e., its output describes the probability density function

$$q(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{K} \pi_i \psi_i(\boldsymbol{y}|\boldsymbol{x};\theta_i),$$
$$\text{s.t.} \sum_{i=1}^{K} \pi_i = 1$$

where K represents the number of components in the model, π_i represents the mixture coefficient of component i, and $\psi_i(\boldsymbol{y}|\boldsymbol{x};\theta_i)$ is the parametric probability distribution corresponding to component i, parameterized by θ_i. With minimal assumptions, a mixture distribution is able to approximate any full conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ arbitrarily well [13].

In our experiment, each von Mises-Fisher component, $\mathrm{vMF}_i(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\mu}_i,\kappa_i)$, could be viewed as a single hypothesis of a trajectory $\boldsymbol{x}$'s true destination, $\boldsymbol{y}$; with a 3-dimensional mean, $\boldsymbol{\mu}_i$, representing its center in Earth-centered-Earth-fixed (ECEF) coordinates; a scalar concentration parameter, κ_i, representing its spread (or variance) over the surface of the Earth; and a scalar weight, $0 < \pi_i < 1$, representing the model's confidence in that particular hypothesis. The model could be configured to produce any finite number of hypotheses for a given input; for our experiment, 20 components appeared sufficient.

The input to our network was a random-length subtrajectory, carefully chosen to give the network no hint of which part of the original trajectory was being sampled. Shorter subtrajectories make up the majority of the set of all possible subtrajectories from a trajectory of length l. In particular, if a subtrajectory is chosen from this set uniformly at random, then the length $\tilde{l}$ of the subtrajectory should follow a triangular distribution with a left bound of 0, a mode of 0, and a right bound of l:

$$\tilde{l} \sim T(0,0,l)$$

First, the subtrajectory's length was sampled from this distribution, and then a valid starting point for the subtrajectory was chosen uniformly at random. In this way, we ensured that we did not over-represent longer, more informative subtrajectories in the training data; such over-representation would lead to an excessively optimistic training loss.

We used a simple neural network architecture based on [8]; this network read the GPS locations of just the first and last 5 waypoints of a sub-trajectory (with repeated padding added as necessary) and then passed that information through 2 fully connected hidden layers consisting of 100 neurons each. Even when trained on just a fraction of a single day's data, this simple model achieved a median top-20 test accuracy of 107.0 km from the true destination for trajectories across the globe. For each test trajectory, this accuracy was calculated as the minimum Haversine distance between the true location, $\boldsymbol{y}$, and the 20 predicted means:

$$\min_{i \leq K} \text{Haversine}(\boldsymbol{y}, \boldsymbol{\mu}_i)$$

An example of this network's output is shown in Figure 12. We expect that by utilizing more data, as well as a more complex model (such as an LSTM or Transformer network), we would be able to achieve even better accuracy.

5 Conclusion of Work and Impact on Customer Goals

5.1 Prediction

Our algorithm is more adaptable and explainable than other trajectory prediction techniques. The alignment algorithm allows us to easily extend our prediction algorithm to predict origin/destination as well as position for the next x minutes into the future for a given observed trajectory. In terms of explainability, the list of origin/destination pair predictions suggested by the algorithm are the origin/destination pairs of historical trajectories who have traveled paths near the observed trajectory in question. And, in the position prediction case, the predicted positions in x minutes are exactly the positions of historical trajectories who have traveled paths near the observed trajectory in question in x minutes.

5.2 Motion Event Definitions

While we believe developing a set of definitions is possible, we conclude that creating a single set of definitions to cover all possible vehicles is not. Additionally, without community input, adoption of any set of definitions will also be very difficult. However, given intelligent categorization and engagement the track community, we believe a set of definitions that will satisfy most use cases is within reach. We have developed a plan to make this happen and believe it will set the stage for the community as a whole.

6 Visions for Future Work

As part of the work done in support of the original Statement of Work, we discovered a number of potential interesting technical problems and applications that could be examined in a deeper way. We only spent a small amount of time considering the problem of ranking the matched trajectories in a meaningful way, and this was entirely related to the quality of their alignment. There are many other aspects of the match that could also be used, such as velocity profiles or day of the week, just to name two. The problem of assigning rigorous probabilities to the predictions was also only examined briefly and would be an interesting problem to systematically study. A different direction for future research would be to implement this in an architecture that corresponded more closely to the production architectures that are currently in use and demonstrate its abilities there. It is probably worth examining how these results would be presented to a typical user and study the most effective ways to present this information.

Another direction for future research would be to understand how to do this in a streaming approach efficiently. In a streaming implementation, the question of how soon one might be able to make a confident prediction could be relevant. This would require understanding the important points in the potential future trajectories where there is a significant divergence between different sets of trajectories.

Finally, given the success of our proof-of-concept neural approach, a third direction for future research would be to explore more complex neural network architectures for our model. Specifically, three architectures hold promise for this domain: convolutional (CNN) networks should be able to pick up on small-scale movement patterns and use this information to inform their prediction; LSTM networks are well equipped to handle time series data, and should be able to gradually refine their predictions as they process each waypoint of a subtrajectory; and finally, the Transformer network is adept at making associations between distant regions of a sequence and may be able to produce the most nuanced predictions of all three of these architectures.

7 Acknowledgements

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This report has been approved for Unclassified Unlimited Release, SAND2021-xxxx R.

References

[1] Neto, Souza Baptista, and Campelo. Combining markov model and prediction by partial matching compression technique for route and destination prediction. *Knowledge-Based Systems*, 154:81–92, 2018.

[2] Xing Wang, Xinhua Jiang, Lifei Chen, and Yi Wu. Kvlmm: A trajectory prediction method based on a variable-order markov model with kernel smoothing. *IEEE Access*, 6:25200–25208, 2018.

[3] Yassine Lassoued, Julien Monteil, Yingqi Gu, Giovanni Russo, Robert Shorten, and Martin Mevissen. A hidden markov model for route and destination prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[4] Zhang and Mahadevan. Bayesian neural networks for flight trajectory prediction and safety assessment. *Decision Support Systems*, 131:113–246, 2020.

[5] Shi, Xu, Pan, Yan, and Zhang. Lstm-based flight trajectory prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[6] Ebel, Göl, Lingenfelder, and Vogelsang. Destination prediction based on partial trajectory data. *arXiv preprint arXiv:2004.07473*, 2020.

[7] Endo, Nishida, Toda, and Sawada. Predicting destinations from partial trajectories using recurrent neural network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172. Springer, 2017.

[8] Alexandre De Brébisson, Étienne Simon, Alex Auvolat, Pascal Vincent, and Yoshua Bengio. Artificial neural networks applied to taxi destination prediction. *arXiv preprint arXiv:1508.00021*, 2015.

[9] Athanasios Tsiligkaridis, Jing Zhang, Hiroshi Taguchi, and Daniel Nikovski. Personalized destination prediction using transformers in a contextless data setting. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[10] Chengyang Qian, Ruqiao Jiang, Yi Long, Qi Zhang, Muxian Li, and Ling Zhang. Vehicle trajectory modelling with consideration of distant neighbouring dependencies for destination prediction. *International Journal of Geographical Information Science*, 33(10):2011–2032, 2019.

[11] Froehlich and Krumm. Route prediction from trip observations. Technical report, SAE Technical Paper, 2008.

[12] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, 1984.

[13] Christopher M Bishop. Mixture density networks. 1994.