Unsupervised Learning Report

Skyler Ebelt
Georgia Institute of Technology
GT ID: 904077010

I. AI Use Statement

In order to complete this project, AI was utilized in a few ways. I only used Claude Sonnet 4.5 for the duration of this project. Claude was utilized for organization, to better manage my workflow on this project. Additionally, Claude was used to refactor broken code segments. I worked to understand every change suggested by Claude before implementing it in my code.

II. Introduction

A. Datasets Utilized

For this assignment we once again are using the same two datasets. These are comprised of the Hotel booking dataset [1] and the Accidents dataset [2].

The hotel booking dataset provides observations into instances of hotel booking, which for these assignments, have been labeled as either canceled or not canceled. The goal of using this dataset with this selected target is to attempt to reasonably predict which observations will result in cancellation. Real world use cases of such predictions would be very useful in formulating advertisement campaigns as well as promotions to retain accounts which seem likely of being canceled. It can be expected that high predictability within this example could strongly contribute to a hotel in maintaining loyal clientele and preventing loss of revenue. This data features about 120,000 records with 32 features. In past experiments, this feature space has done an adequate job in producing models capable of producing some predictability and allowing for certain optimization techniques to modestly further generalize over baseline performance.

In the case of the Accidents data, observations are labeled with a severity score. Severity is a multiclass variable with larger severity scores corresponding to more dangerous or lethal accidents, with smaller scores corresponding to, for lack of a better term, less severe accidents. This data would be highly applicable from the perspective of law enforcement, or departments overseeing highway/road safety. By understanding how certain conditions produce certain severity scores. This can guide decision making within roadway maintenance and planning. With approximately 7,000,000 rows and 46 features this data showcases a rich feature space producing models which show strong predictivity, and allowing for optimization to produce decent improvements over baseline performance. Because of this, I feel performance of models and techniques utilizing this data will be much higher than that of those on the hotel data.

B. Hypothesis

This analysis was done to examine and measure the relationship between dimensionality, and clustering within neural network classifiers. The same data which was utilized the prior two assignments are to found here once again. Both datasets present unique real world observations, which both capture trends, feature spaces, dimensionality, etc. which greatly vary. Due to this, key differences in the performance of models and techniques can be seen depending on which data is utilized for any given task. Because of this I hypothesize that the utilization of dimensionality reduction will vary in impact within the performance of clustering algorithms. We should expect to see variance in model performance due to data specific patterns. Furthermore, it should seem intuitive that we see the largest performance increases within accidents data due to a larger amount of observations but also a feature space which may be more indicative of actual patterns (for instance it should be expected that a weather pattern of rainy and a time of night to strongly correlate to an increased severity score). Hotel booking however seems to be a more difficult task to predict upon due to a smaller view of the conditions the party booking a hotel stay is experiencing. Because of this, we can expect larger accuracy scores and performance increases within predicting severity within accidents. We should expect slight performance increases within hotel cancellation predictions. Additionally, we can assume that utilizing cluster derived features will improve model performance [4].

C. Objectives

The objective of this project is to measure how dimensionality reduction affects performance within unsupervised learning techniques. This should provide a view into what steps to take and when, and which methodologies to employ within the unsupervised learning/clustering space. This should serve as an artifact for reference when considering an approach to a problem well suited for the employment of a unsupervised learner.

III. Data Preprocessing & Hardware

A. Data Preprocessing

Hotel data required imputation to accommodate for NA values. The children, agent and company columns received 0 where data was not known, and country received 'unknown' when no known country was on file. Duplicate observations were handled by removing the duplicates. Label encoding was used on categorical variables in order to cater towards

computational feasibility. 27 features were kept to be further explored in the future analyses.

The accidents dataset was limited to 1,000,000 observations for the purposes of these experiments. In order to work within PyTorches framework, severity scores had to be recoded from [1,...,4] to [0,...,3]. 42 out of 45 features were selected to move forward with experimentation. Both datasets were standardized using StandardScalar in order to prevent leakage and ensure consistency between datasets for a fair comparison.

B. Evaluation & Hardware

This assignment did not require further sampling beyond the posted limits. All hotel observations were considered while accident data was capped at 1,000,000 records as instructed in the report guidelines. This code was able to execute in under an hour, a huge improvement from past assignments which spanned multiple hours on average just to run certain sections of the analysis. Again, I have an i7-9700 with 32 GB of ram as well as an RTX 2080 super. I once again utilized CUDA tensors so as to maintain faster runtimes. I used the same convention for establishing seeds as can be seen in the first two assignments, associating them with my GTID (4077010). This was done using Numpys random seed functionality but also within PyTorches manual seed functionality.

All models used PyTorch and Scikit-learn for implementation and Seaborn for figure creation. Once again, CUDA tensors were employed via PyTorch to enhance runtimes which proved to be successful in my specific circumstance. For evaluation and comparison, metrics such as silhouette, Davies-Balduin, and Calinski-Harabasz scores were used as suggested in the project guidelines as well as BIC/AIC where applicable.

IV. Clustering (Raw Data)

A. Implementation

This first study involved initializing clustering algorithms trained on raw data (Hotel, and Accidents). This was done in order to establish a baseline metric of model performance to better describe and depict data specific patterns/performance variation between datasets, and performance variation between techniques in the following studies. In order to quickly implement this baseline, little code is needed. PyTorch's clustering functionality was utilized to create these clusters and Seaborn was used to plot the corresponding output. The data used to train the clustering models were preprocessed but otherwise unmodified. No other techniques were utilized to reduce dimensionality, so as to provide a baseline to show improvement or the lack thereof in future experiments.

The hotel dataset for these training purposes, contained 27 features and established K-Means and GMM with K=3 components. The entirety of observations was used within training and prediction. Accidents data received similar treatment, K was set to equal 4 however within K Means and GMM. Accidents also required training using the full dataset, but for metric calculation a stratified sample in order to maintain population proportions but prevent huge computational costs. Due to the large amount of observations (even with limiting them to 1,000,000) calculating metrics utilizing every data point would be highly exhaustive. This succeeded in providing descriptive metric evaluations (plots) without excessive run times.

B. Results & Analysis

TABLE I

Clustering Performance on Hotel Booking Dataset (k=3)

Metric	K-Means	GMM
Silhouette Score	0.0937	0.0537
Davies-Bouldin Index	2.7640	3.2967
Calinski-Harabasz Score	5458.13	3628.33
BIC	—	−902,152.13
AIC	—	−913,293.82

For hotel data K-Means and GMM were evaluated to compare model performance. Both instances displayed best performance at K = 3. Table II shows low silhouette scores, suggesting clusters not easily linearly seperable. Despite this Calinski-Harabasz (CH) scores shows that centroids are well placed, capturing strong K-Means. Davies-Bouldin (DB) index indicates that K-means performs better with separating clusters in this specific scenario.

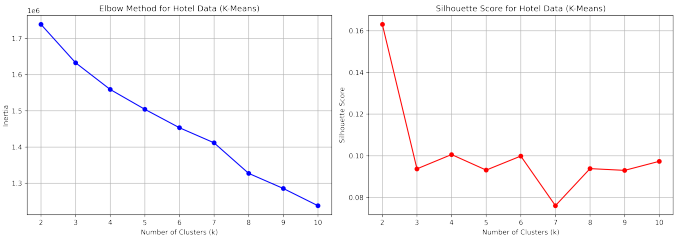

Fig. 1. Hotel K-Means Scores

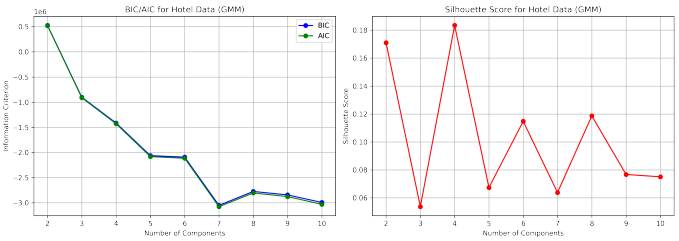

Fig. 2. Hotel GMM Scores

TABLE II

Clustering Performance on US Accidents Dataset (k=4)

Metric	K-Means	GMM
Silhouette Score	0.2392	0.2706
Davies-Bouldin Index	1.7292	2.0587
Calinski-Harabasz Score	5229.16	4448.56
BIC	—	−2,669,161.93
AIC	—	−2,681,536.08

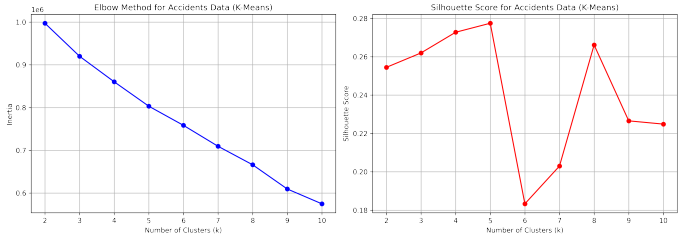

Fig. 3. Hotel K-Means Scores

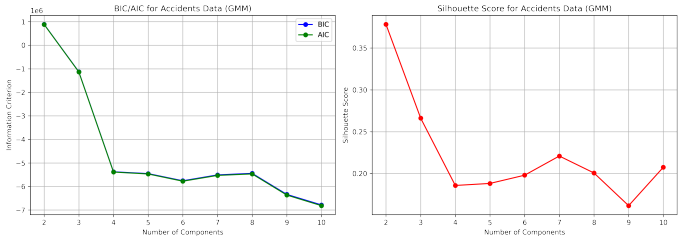

Fig. 4. Hotel GMM Scores

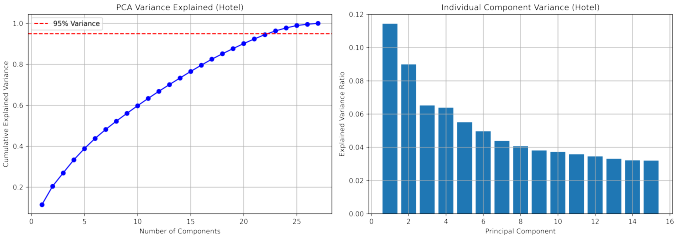

Fig. 5. Hotel PCA Variance Analysis

For the Accidents model, a sample of 50,000 was used in metric calculation to ensure reasonable computation time. The Accidents data performed much better in raw clustering, with similar performance spanning both K-Means and GMM. High silhouette scores desribes easily seperable clusters. Evaluating the DB index found in table II we can see that K-Means performed better here creating tighter seperated clusters.

V. Dimensionality Reduction

A. Implementation

This experiment required developing three linear dimensionality reduction (DR) methods. This was done in order to establish methods of DR for future testing within dimension reduced spaces and the effect it will have on unsupervised learning techniques. The goal of such methods is to determine feature importance, allowing for dropping of less relevant data. Effectively, these methods drop features to limit dimensionality while maximizing variance in the data [3]. This suggests supporting lower computational constraints by removing unnecessary features/dimensionality while minimizing impact on predictivity of models working within DR spaces (by allowing more relevant features to remain unchanged). Furthermore, this required fitting DR models on processed training data, transforming training and test splits, and then outputting the results via plots and print functions. The DR methods were comprised of PCA, ICA and Random Projection.

In order to establish PCA, scikit-learns PCA was used without any component restriction, n_components was set to equal 23 in the case of hotel data and 19 for accidents.

For ICA, fast ICA was used with the same n_component values found in PCA. 500 iterations were allowed so as to attempt to guarantee convergence. Kurtosis was calculated using Scipys kurtosis functionality.

Finally, random projection was implemented with GaussianRandomProjection from sklearn. n_component values once again matched the previously stated values which can be found in the PCA and ICA experiments. Five instances of this method were established and sampled, from here Pearson correlation was calculated as well as distance, which was used to calculate error.

B. Results & Analysis

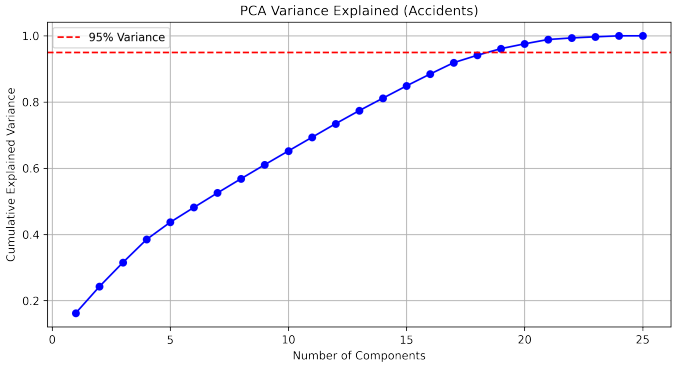

Fig. 6. Accidents PCA Variance Analysis

Hotel dimenstionality reduction shows strong variance between each initialization of each technique. PCA found that at 23 features, 95% of variance could be accounted for. ICA selected 23 variables with a mean absolute kurtosis of 306.5 and finally RP selected 19 features with a mean absolute kurtosis of 1972.48. Accidents data similar to hotels, found that 19 features accounted for about 95% of variance, ICA selected 19 features with a mean absolute kurtosis of 1972.5 and RP selected 23. Strong compression in the case of accidents suggests a theme of feature redundancy.

VI. Clustering in DR spaces

A. Implementation

This experiment utilized the methods in both part 1 & 2 of this report. Utilizing both DR and clustering. Clustering raw data provided a baseline of performance without any changes to dimensionality. Meanwhile, experimenting with DR in the previous testing provides insight into how each method transforms the feature space to further support analysis in these experiments. This implementation used all three DR methods and used the same strategy for clustering as can be

found in part 1. This ensures a fair comparison of each DR method and how it compares to initial clustering performance per dataset.

Clustering methods were initialized using the same parameters as can be found in the first experiment. Models were fit on the full (but reduced) datasets for both hotels and accidents. In other words, all observations are considered (bear in mind for the purposes of this assignment, the Accidents data has been limited as outlined in the project guidelines) features of course have been dropped as a result of the DR methods. For evaluation Hotel models used the entire dataset for calculations, but again Accident models used a sample when calculation metrics to prevent computational limitations from arising. Similar to previous results, the results of this strategy depict models performance well without being exhaustive. The metrics as stated previously were used in determining model performance.

B. Results & Analysis

TABLE III
CLUSTERING PERFORMANCE IN DR SPACES: HOTEL BOOKING DATASET

DR Method	K-Means		GMM	
	Silhouette	Davies-Bouldin	Silhouette	Davies-Bouldin
PCA	0.0988	2.6918	0.0561	3.2271
ICA	0.0536	3.4799	0.0374	3.8506
RP	**0.1227**	**2.4210**	**0.1838**	**3.1220**

DR produced scattered results on the Hotel dataset which was proposed by my initial hypothesis. Seemingly, table III supports that the accidents data may present more noise, and less trends within it making prediction harder, the results of this analysis further support this idea. GMM degraded model performance, but K-means slightly improved it, with RP scoring highest within this analysis. This might suggest that the reduction in noise, working in conjunction with RP was able to produce some slight performance increases due to its tendency to preserve distance. PCA and ICA failed to meaningfully increase model performance, with ICA degrading performance. RP was able to boost performance but minimally.

TABLE IV
CLUSTERING PERFORMANCE IN DR SPACES: US ACCIDENTS DATASET

DR Method	K-Means		GMM	
	Silhouette	Davies-Bouldin	Silhouette	Davies-Bouldin
PCA	**0.2808**	**1.6306**	0.2076	2.8408
ICA	0.2585	1.7067	0.2390	2.5595
RP	0.2635	**1.1594**	0.1656	2.7659

Considering table IV, we can see that K-Means using PCA outperformed all other methods, followed shortly thereafter by RP and then ICA. With decent improvements over baseline, and strong DB scores this indicates the DR methods succeeded in reducing noise present with the accidents data while preserving key trends within the data. DB scores indicate clusters which are easily separable. This supports my hypothesis which suspected that accidents data would perform better, due to a larger access to observations and evidence of less noise within data in previous assignments.

VII. NEURAL NETWORKS IN DR SPACES

A. Implementation

This analysis involved training a neural network on both the original data after being exposed to dimension reduction methods. This was done so as to examine the performance of the baseline NN instance as it compares to the NN instance on each clustering algorithm. To do this data was split into an 80% training partition and a 20% testing partition before once again splitting into another 90% training split and 10% testing split. All splits were done using the same random seed convention as previously stated. This strategy was employed to ensure similar to same conditions in training testing and validation splits. Four instances of NN models were implemented, first the baseline NN, then NN with PCA reduced dimensionality, next NN with ICA reduced dimensionality, and finally the NN trained on Gaussian Random Projection reduced space.

Each instance of testing had identical conditions, each NN model had an architecture of four hidden layers comprised of 150, 100, 75, and 50 neurons respectively. These NN models utilized ReLU activation. Models were allowed 50 epochs to converge with a patience of 5 epochs. Batch size was established as 512 and a learning rate of 0.01 was utilized. Performance was measured using accuracy and F1 scores.

B. Results & Analysis

TABLE V
NEURAL NETWORK PERFORMANCE ON DR DATA (US ACCIDENTS)

Input	Acc.	F1	Time (s)	Ep.	Dims	Val Loss
Original	**0.7983**	0.7288	**96.74**	**10**	25	**0.4969**
PCA	0.7919	0.7153	111.48	12	19	0.5106
ICA	0.7981	0.7156	159.73	17	19	0.5108
RP	**0.7983**	**0.7253**	134.39	15	19	0.5120

The results of this analysis show that DR did not meaningfully improve performance. All methods except for RP degraded model performance, with RP actually failing to produce any performance increase it simply remained the same in the case of test accuracy. No model produced better F1 scores, each one degraded the performance of calculating this metric. The reduction of dimensionality seems to cut out key context within the data which the NN models rely upon to properly generalize. Without access to this data we can clearly see that model performance fails to outperform the baseline in any regard.

considering the validation lost curves we see further evidence that the DR methods cut out information key to the NN models ability to learn. With this specific architechure and data, the full list of features seems to be needed to properly model this data.

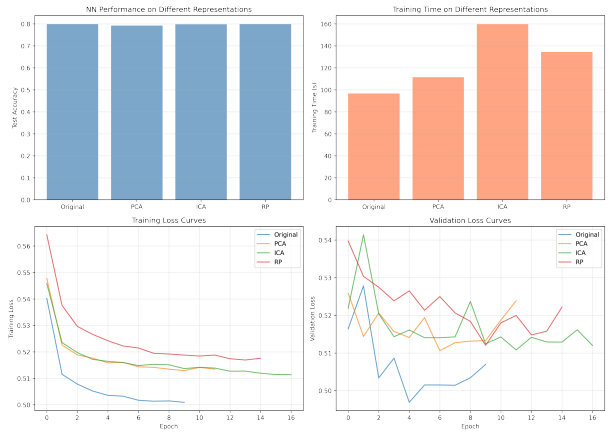

Fig. 7. NN DR Comparison

VIII. Neural Networks on Cluster Derived Features

A. Implementation

This final experiment required training NN models on features derived from clustering algorithms. By using cluster derived features, the goal is to enhance supervised learning techniques. By deriving features, clustering algorhithms may produce more relevant features which reduce noise increasing models performance [4]. Standard supervised learning techniques, trained on non-cluster derived features can be beat out by supervised models trained on cluster derived features [4]. Experiment 1 clustering methods were applied to only the Accidents data set using the same parameters in the first analysis. One hot encoding was utilized on extracted features. The same conditions are provided for the NN models as before, architechture and parameters remain unchanged to further the effects of these experiments. From this three conditions were established, the first of which only contained features derived from clustering techniques, the next which used the features already present within the accidents data alongside the derived features, the final condition compares features derived using K-Means versus GMM. These conditions seek to outline a few key points. The first of which is the effect of using supervised and unsupervised learning in conjunction. Measuring performance changes with this in mind can illustrate the effect of the cluster derived features. Next, after establishing the effect of both learning methods working together, an idea of how data processing affects performance as well as algorithm specific conditions can effect performance.

B. Results & Analysis

TABLE VI
NN Performance with Cluster-Derived Features

Configuration	Acc.	F1	Time (s)	Dims
Cluster Features Only	0.7967	0.7065	73.44	16
Original + Clusters	0.7917	**0.7350**	170.92	41
K-Means Only	0.7966	0.7066	74.33	8
GMM Only	0.7967	0.7065	**65.67**	**8**

Taking a loot at table VI, All models showed near identical performance within test accuracy with cluster features only (CFO), K-Means, and GMM all being virtually tied (tied within F1 scores as well). While original and clusters (OC) lacked in test score but had a slightly higher F1 score but yielded the longest train time. OC possessed a massive 41 input dimensions, contributing to its long by comparison run time. Scores within test accuracies showcase an inability to generalize much further than already established within benchmark testing, with OC degrading testing accuracy and everything else essentially stalling. F1 scores show that OC did the best job handling class imbalances, better capturing instances of classes which appear less frequently, this however, comes at the cost of a much more computationally expensive model. The K-means model seems to have performed the best in this experiment however, with minimal losses to performance, while accelerating the training time. Scoring well in both test accuracies and F1 scores, this seems to be a valid shortcut when training models with access to large amounts of data.

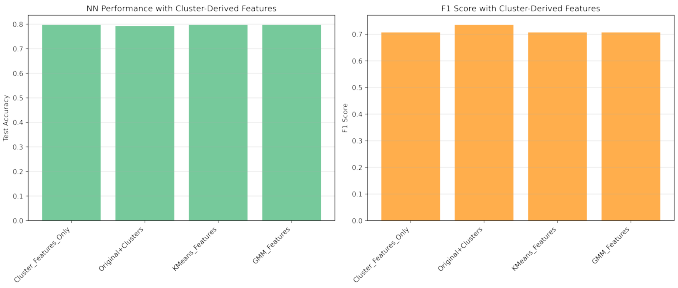

Fig. 8. NN Cluster Comparison

This analysis seems to point solely to the fact that we were unable to increase accuracy. Computation time however, was reduced providing a view into potential use cases for these methods. While these results don't support our hypothesis, we can reasonably conclude that these methods can be employed to save time during training, or to potentially enable machines that are less computationally inclined to initialize these models. Additional testing could also occur to measure how NN architecture effects these results, as well as further tweaking hyperparameters. For the intents of this paper however, this analysis did not provide direct evidence to support our hypothesis.

IX. Conclusion

The results provide mixed evidence for and against the initial hypotheses. The results of this analysis once again show that the feature space (even within dimensionality reduction) seems to capture, and is more descript of patterns within the accident data as compared to the hotel booking data. This same trend can be found within the first two assignments completed for this course. Accidents data, when ingested in each instance of modeling we have done so far, outperforms hotels. The two most likely reasons for this are as follows: Hotel data seems to have too little observations,

whereas accidents data has more than enough for models to reasonably generalize. Additionally, the nature of predicting severity versus hotel booking cancellation, seems to be an easier task to predict. Next, feature space in the accidents data provides key information, with a wealth of information describing conditions before an accident which is reflected by stronger predictive performance within testing. Hotel data seems to fail to provide all relevant conditions, I won't say it is necessarily hard or impossible to predict the outcomes we have looked at within the hotel data across these assignments, but it would seem likely that a good next step in a real world scenario would be to attempt to expand the amount of data we have access to for this task.

As for further discussing these results, the experiments produced output which is not consistent with our initial hypothesis, with one slight caveat. A very mild performance increase can be observed within clustering after utilizing DR methods. Because performance did not meaningfully increase, I would have to say that more testing would be required to fully understand if, when, and how DR can boost performance of clustering methods. Furthermore, it can be seen that utilizing cluster derived features did not improve performance. Again in real world scenarios I would recommend further testing as certain aspects of this implementation may require modification, NN architecture for instance may be a good place to start within further increasing performance. One surprising element of these experiments, and one that may redeem them, is the decrease in computation time within NN training on cluster derived features. Although typical indicators of performance did not show a better generalized model, the time it took to produce a similarly accurate model was reduced. This describes, if nothing else, one strength of these methods. These methods can be used in order to reduce computational load and compute time, which can be incredibly useful especially if models are stored on and used within cloud platforms. This can save time and money, without sacrificing too much performance.

REFERENCES

[1] Hotel Booking Demand: Data: Hotel booking demand (H1/H2), Ant´onio, Almeida & Nunes (2019), Data in Brief 22:41–49, doi:10.1016/j.dib.2018.11.126.

[2] US Accidents: Data: US Accidents (since 2016), Sobhan Moosavi et al., CC BY-NC-SA 4.0, retrieved from Kaggle US Accidents.

[3] Exploring unsupervised feature extraction algorithms: tackling high dimensionality in small datasets. Niu H, McCallum GB, Chang AB, Khan K, Azam S. Sci Rep. 2025 Jul 1;15(1):21973. doi: 10.1038/s41598-025-07725-9. PMID: 40595281; PMCID: PMC12216002.

[4] "Unsupervised Learning and Clustered Connectivity Enhance Reinforcement Learning in Spiking Neural Networks," P. Weidel, R. Duarte, and A. Morrison, Front. Comput. Neurosci., vol. 15, 2021, doi: 10.3389/fncom.2021.543872.