Simultaneous Localization of Multiple GNSS Interference Sources via Neural Networks

Capt David Besson

GPS Directorate

ION GNSS+ 2017

Sept 29, 2017

Portland, Oregon

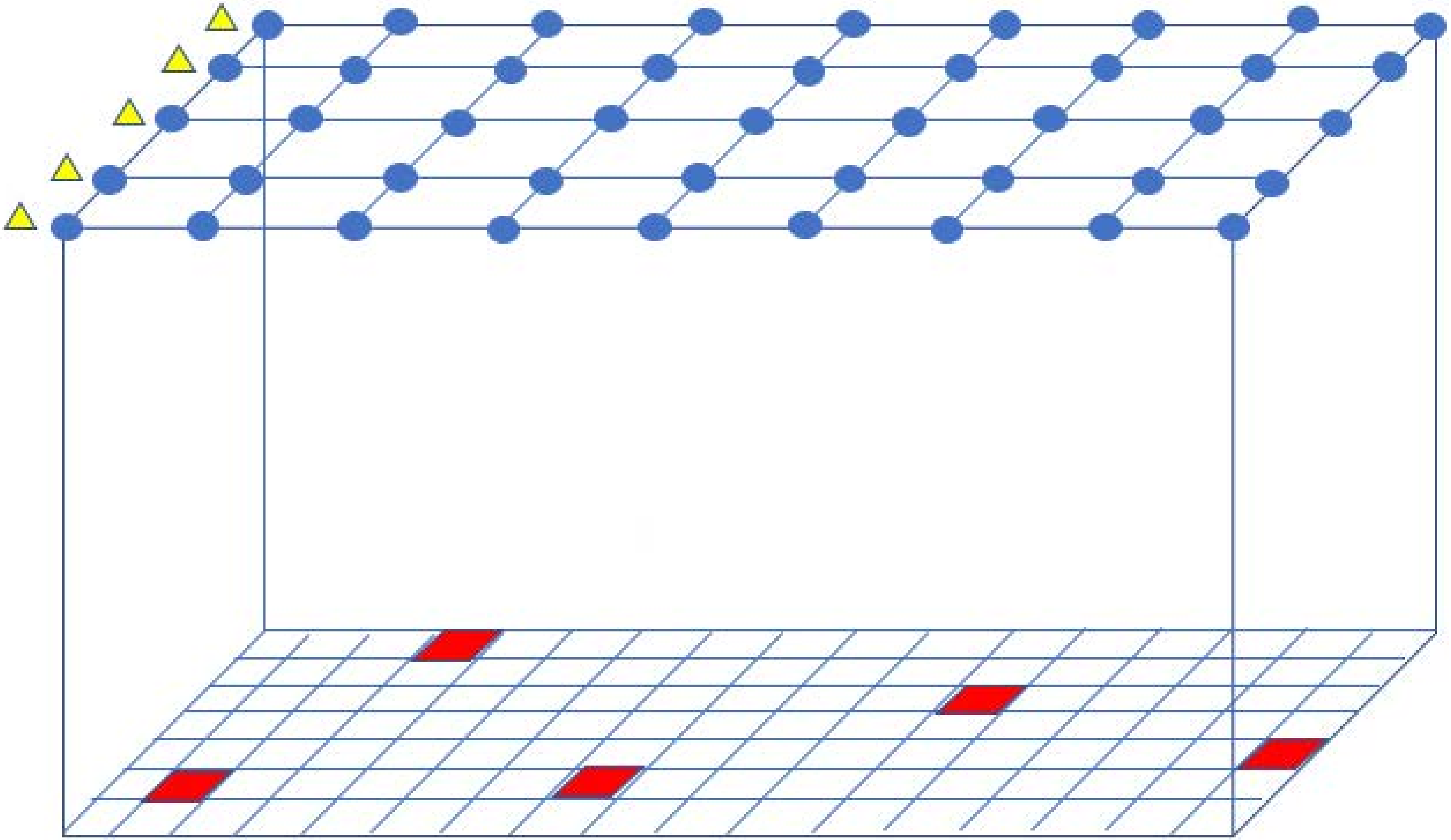
Session D6: GNSS Interference Detection and Localization Algorithms

The views expressed in this presentation are those of the author and do not reflect the official policy or position of the GPS Directorate, United States Air Force, United States Department of Defense, or United States Government.

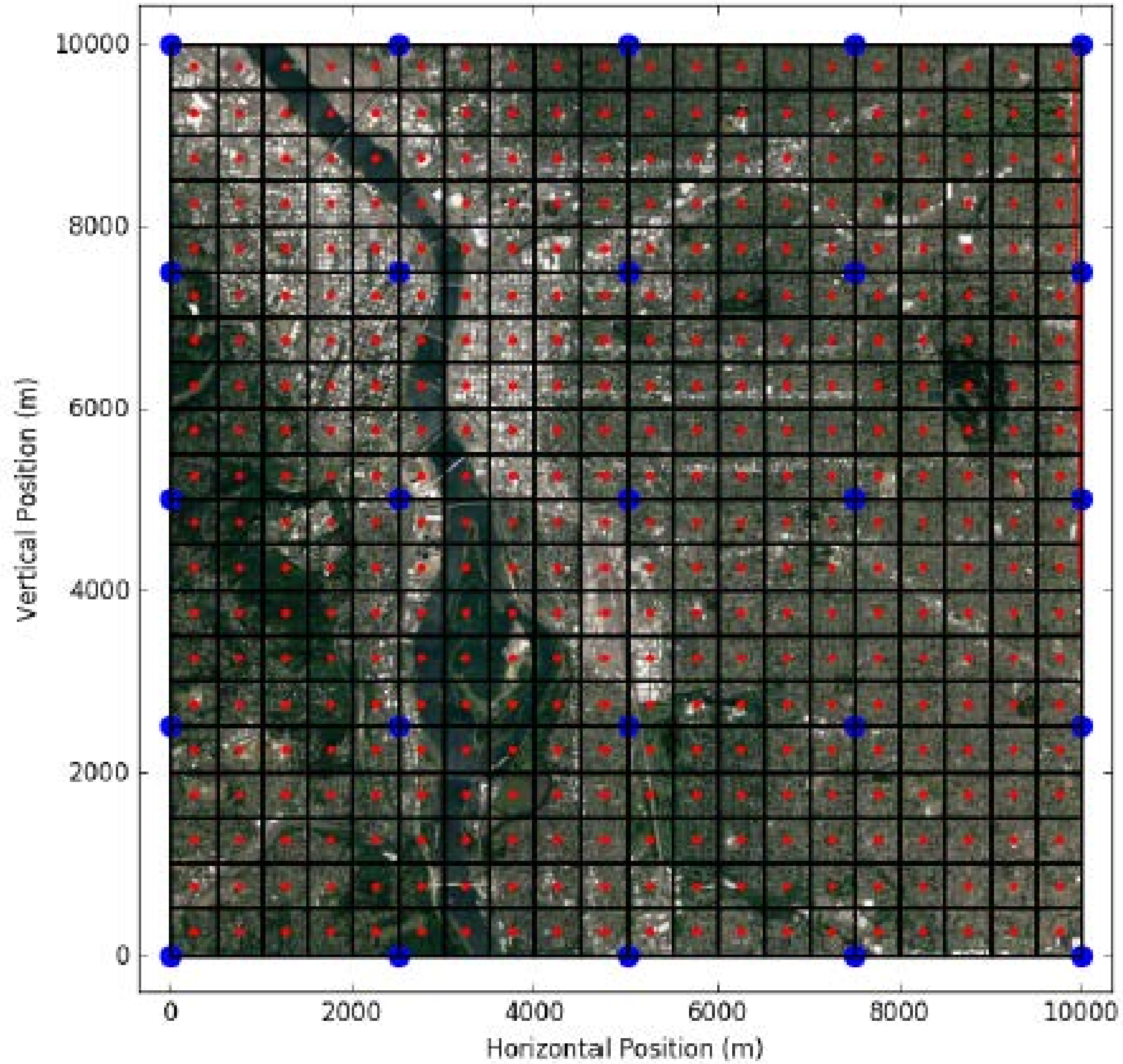
Overview

- Problem statement
- Overview of artificial neural networks and their applications
- Creation of training and test sets
- Performance metrics and analysis
- Inspection of trained neural networks
- Test cases
- Conclusion and Future work

Problem

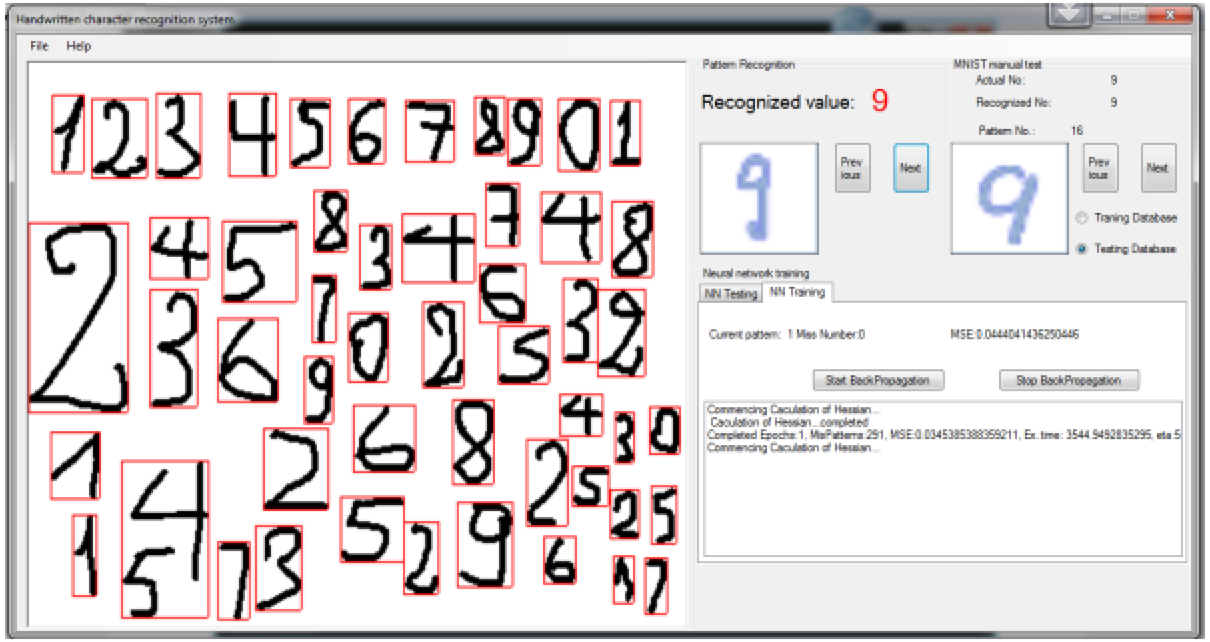


Real-world Context



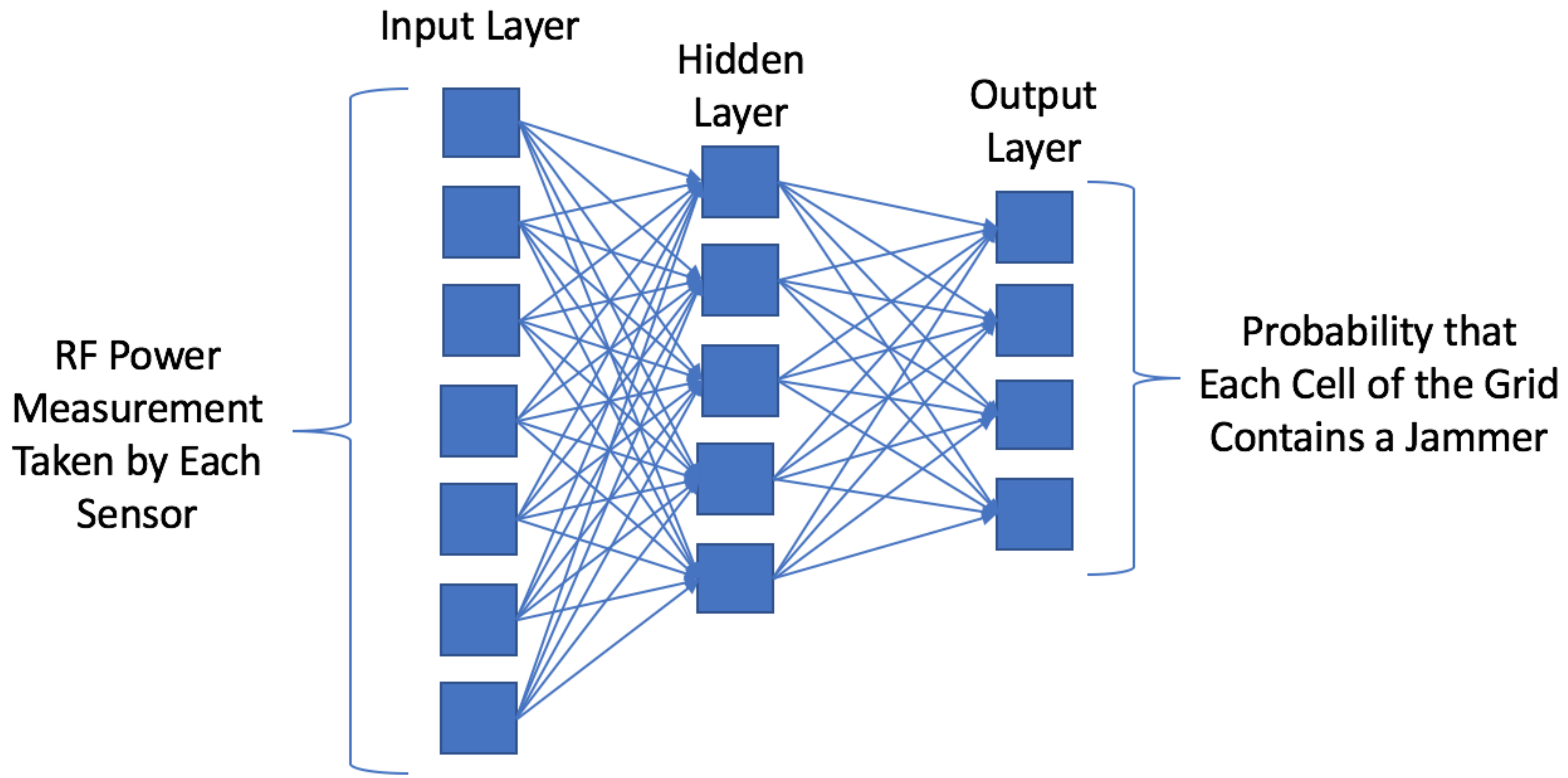
Neural Networks for Optical Character Recognition

https://kuanhoong.files.wordpress.com/2016/01/mnistdigits.gif

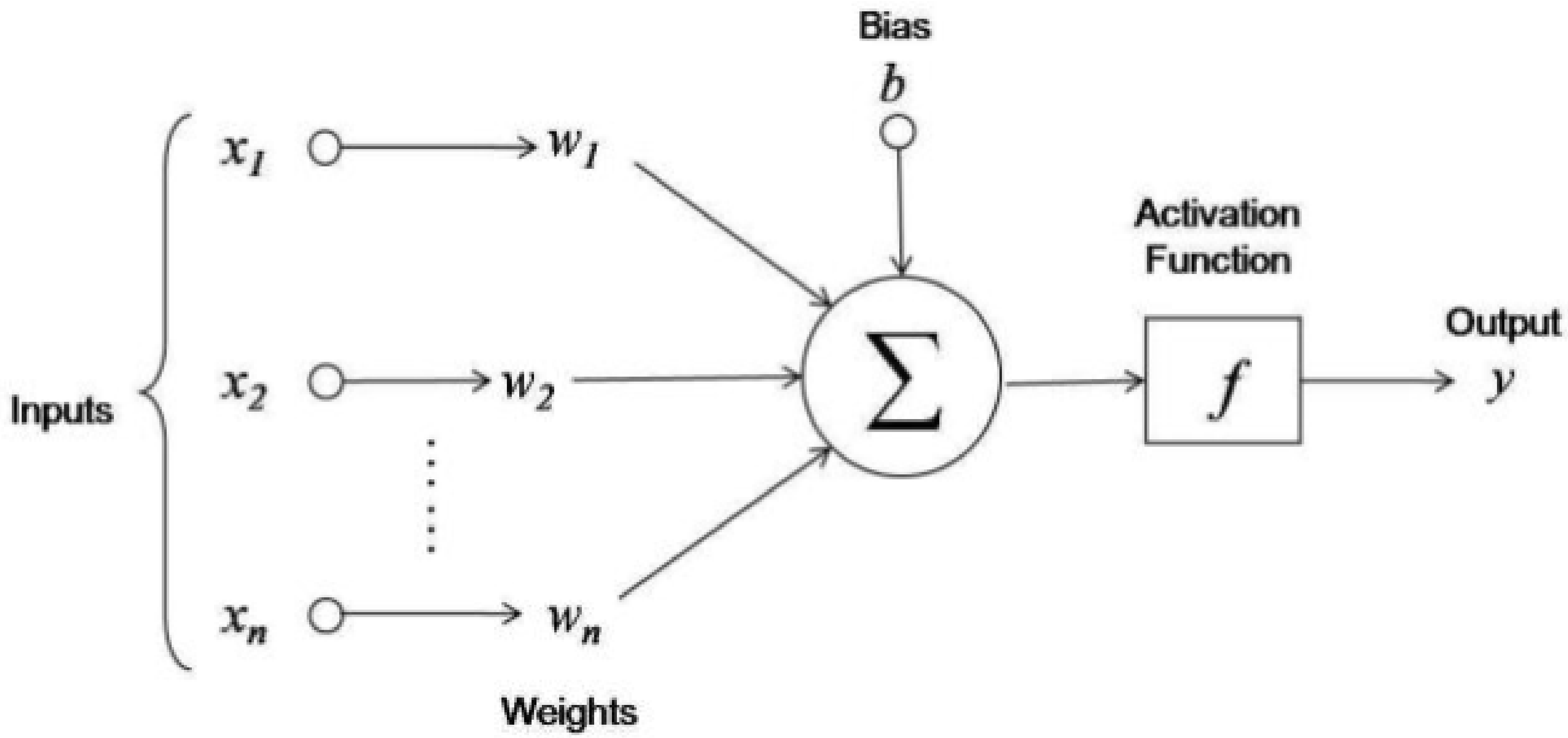


http://www.codeproject.com/KB/recipes/NNHandwrittenCharRecCs/image001_Small.png

Sample Neural Network Structure

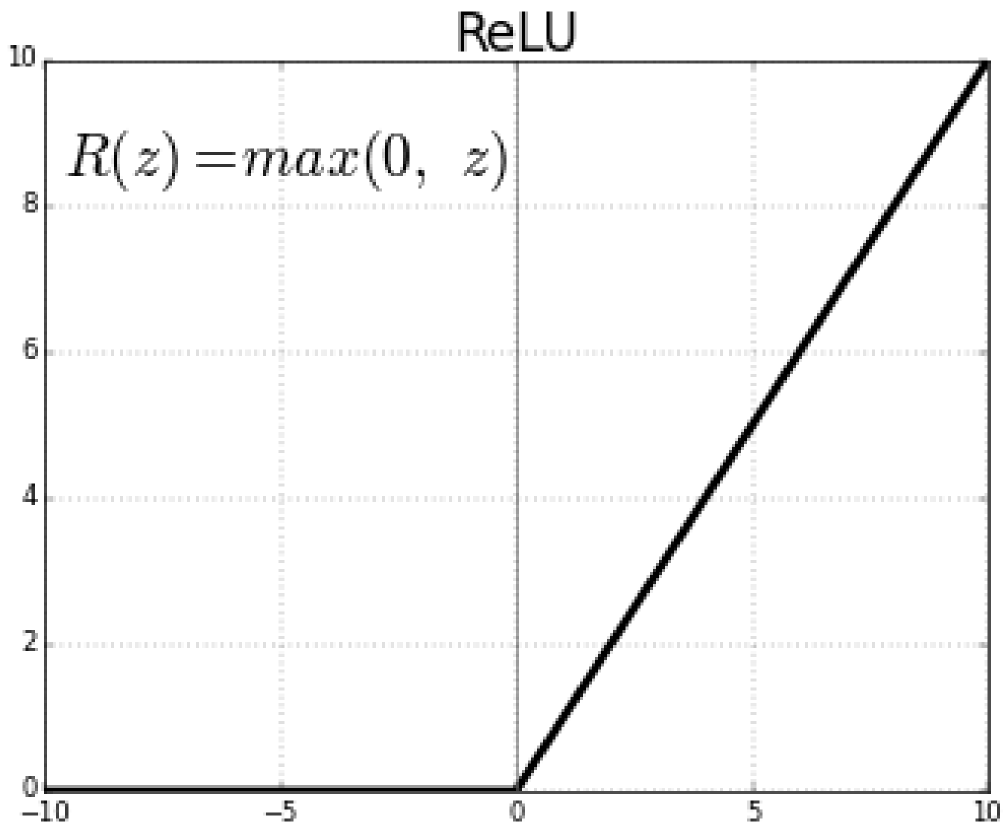


Artificial Neuron

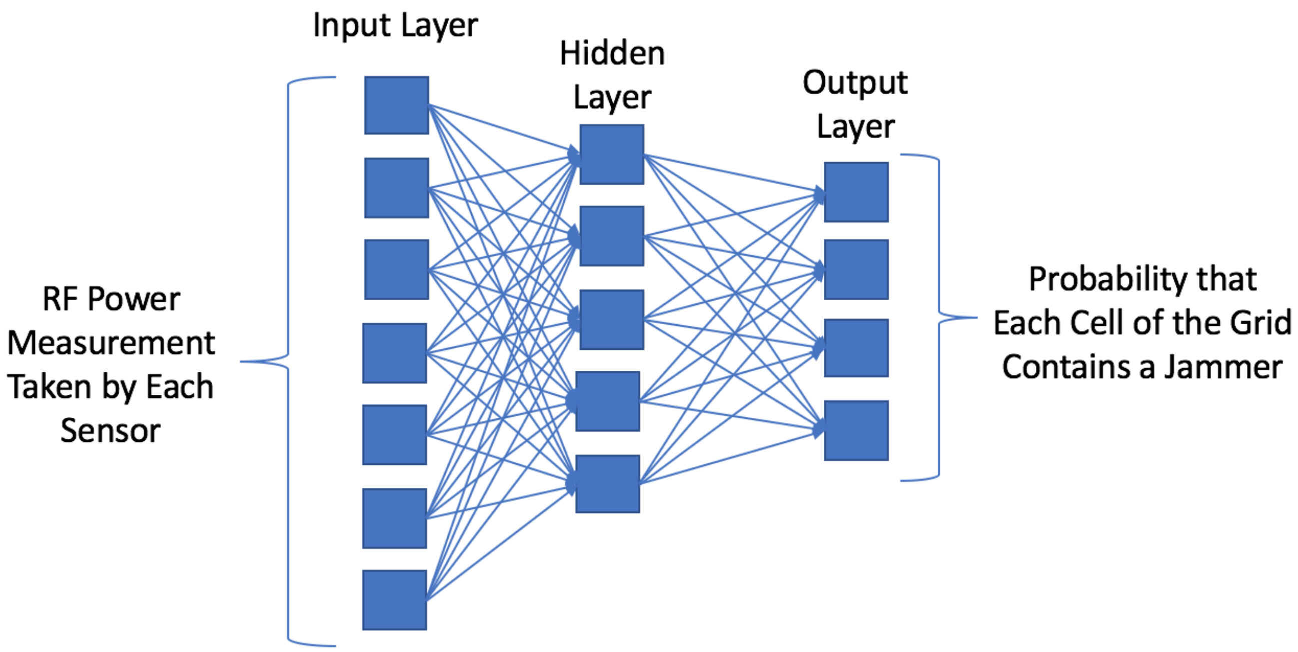


http://blanco.io/blog/machine-learning/neural-networks-and-backpropagation/

Activation Function



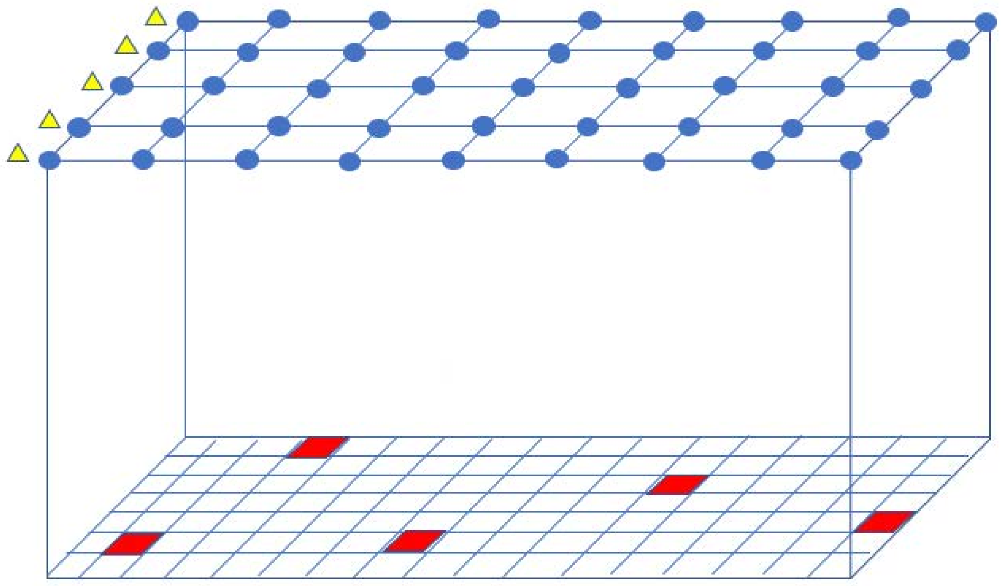
https://ml4a.github.io/images/figures/relu.png



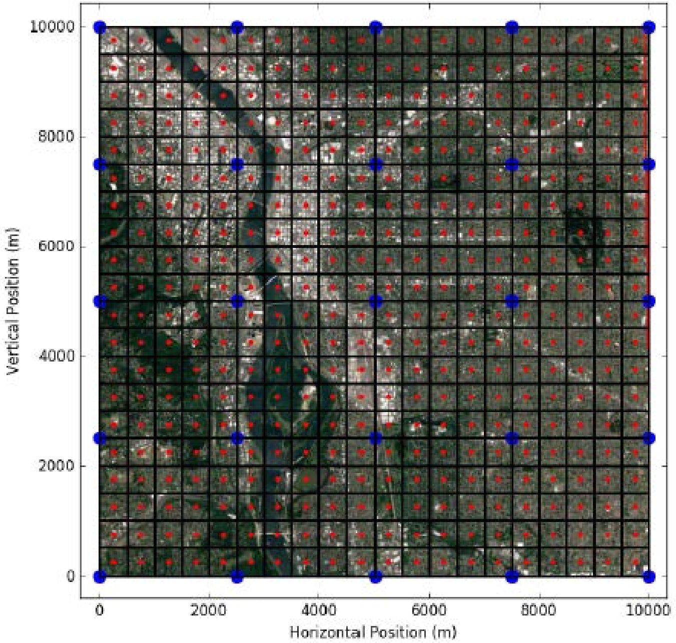
Creation of Training Sets

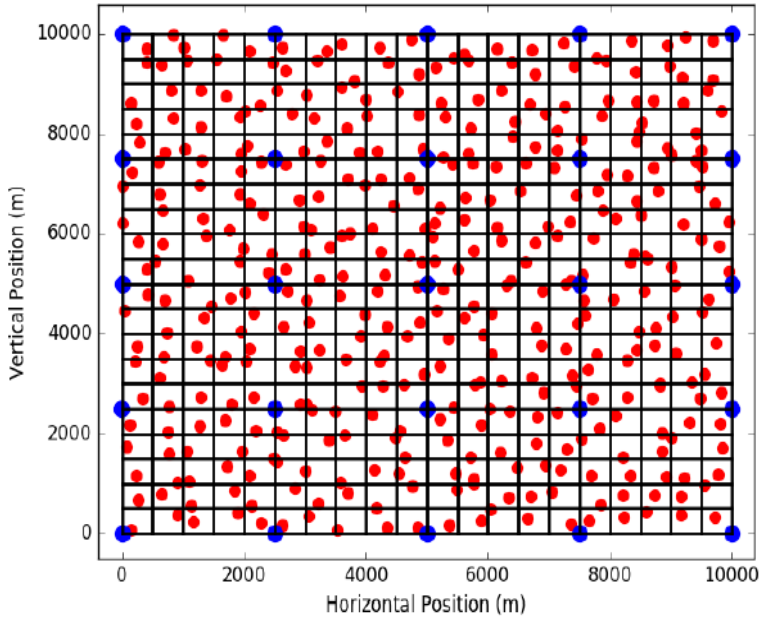
- Goal: Choose a representative sample of sets of training samples that contain different numbers of interference sources.
- Proposed Solution: Randomly select m samples of combinations sets of size [1,…,10] from a total pool of size ${}_{400}C_m$. For those samples, simulate the aggregate power measurements at 25 sensor positions.

m	Total Size of Training Set (n_c)	m	Total Size of Training Set (n_c)
10	490	1000	9400
30	670	3000	27400
100	1300	10000	90400
300	3100	30000	270400



Easy and Hard Training Sets





- Tx Power: 100 W
- Altitude: 0 m
- Position w/in Cell: Centered
- Sensor position error: 0 m

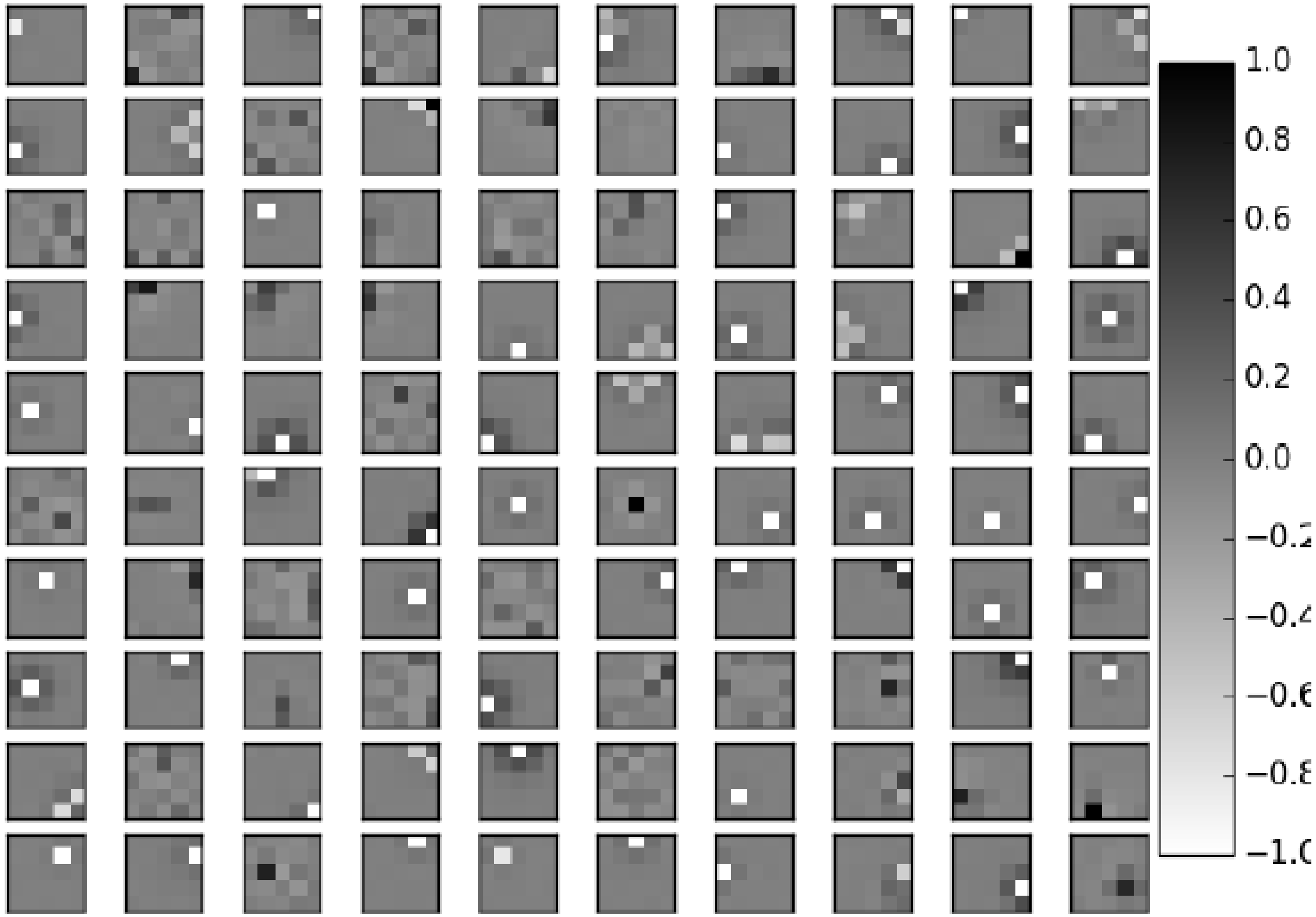
- Tx Power: Random ϵ [50,150] W
- Altitude: Random ϵ [-3,3] m
- Position w/in Cell: Random
- Sensor position error: Random ϵ [-2,2] m

Ok cool, but it’s hard to implement the neural network right?

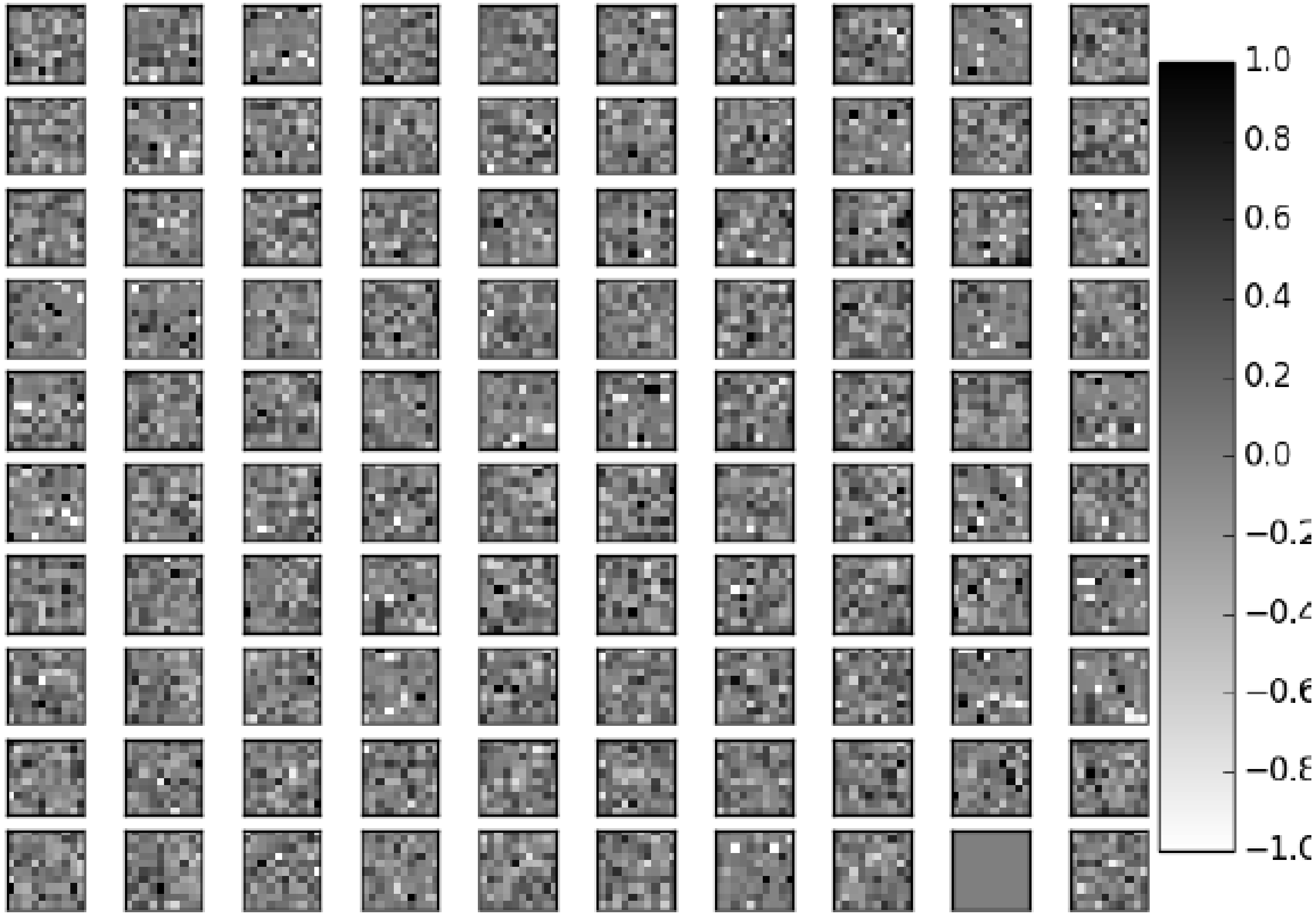
- Not with Python!

```
from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(hidden_layer_sizes=(100, 100, 100,), verbose=10, random_state=1, solver='adam',
    max_iter=1000) #create the pre-trained neural network

mlp.fit(X_train,y_train) #train the neural network with the training set

predictions = mlp.predict(X_test) #predictions are 1 if the probability is 0.5 or higher
predictions_proba = mlp.predict_proba(X_test) #predictions as raw probabilities
predictions_log_proba = mlp.predict_log_proba(X_test) #predictions as probabilities on a log scale
```

First Hidden Layer of Trained Neural Network

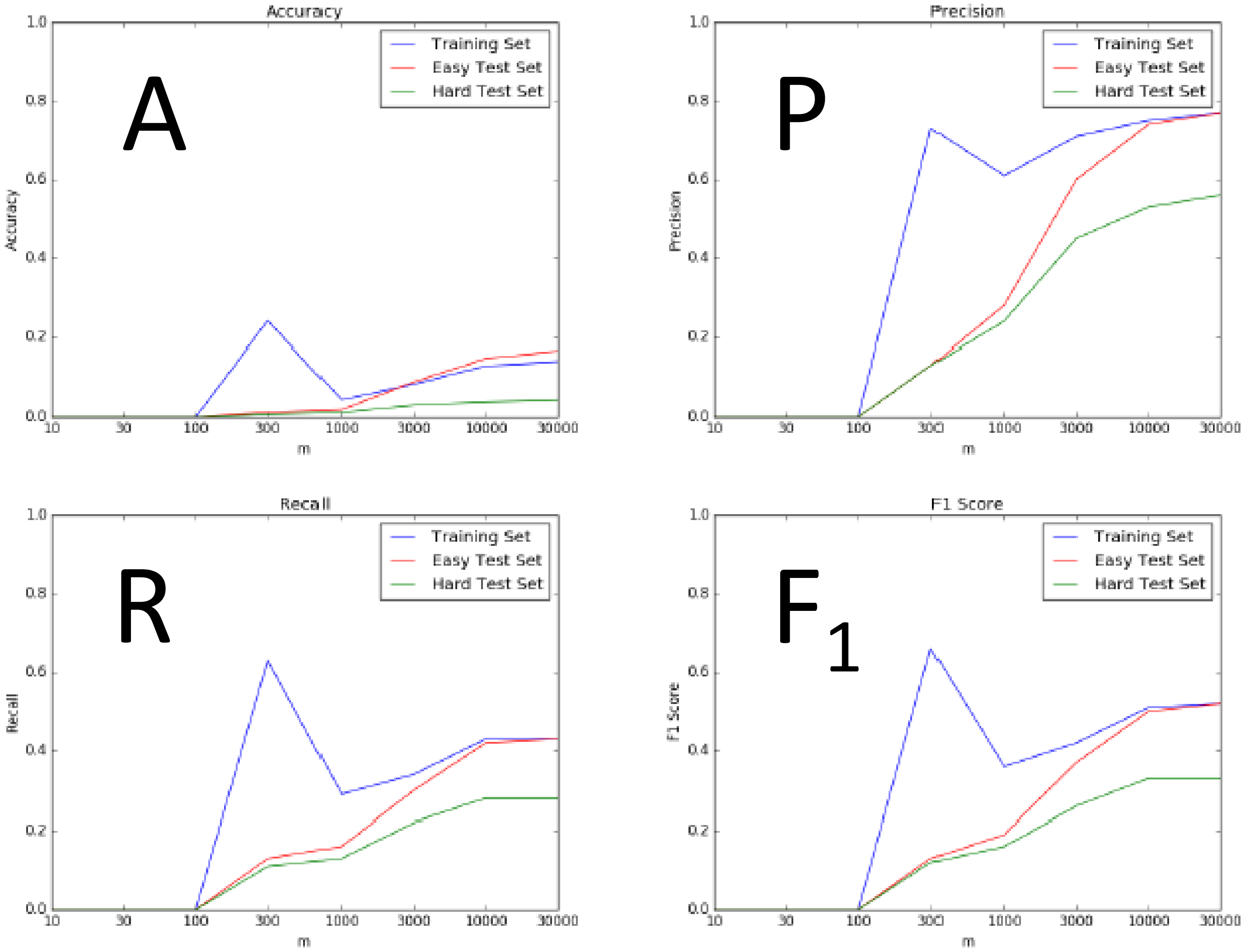


Second Hidden Layer of Trained Neural Network

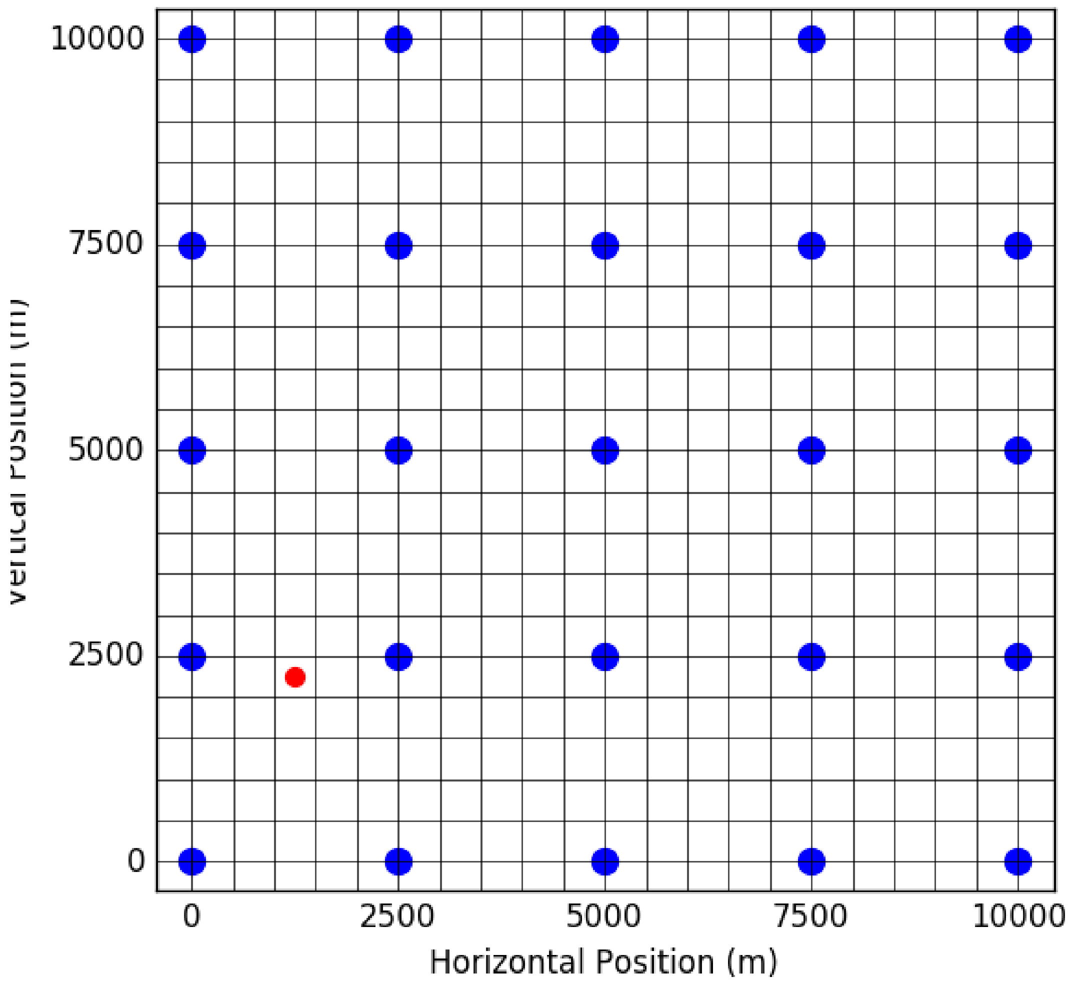


Performance Metrics

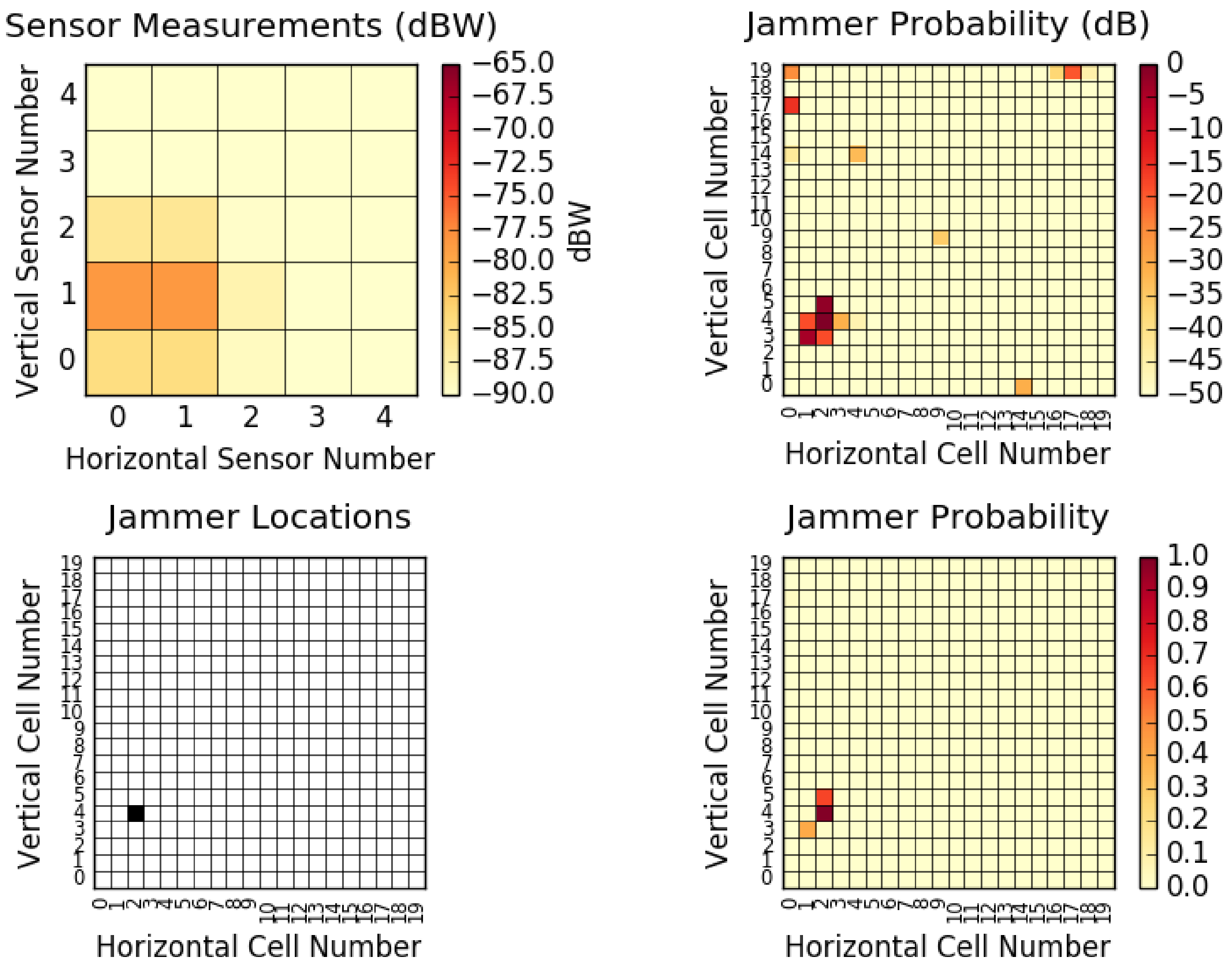
- Accuracy $$A = \frac{T_p + T_n}{T_p + F_p + T_n + F_n}$$
- Precision $$P = \frac{T_p}{T_p + F_p}$$
- Recall $$R = \frac{T_p}{T_p + F_n}$$
- F_1 Score $$F_1 = 2\frac{P \times R}{P + R}$$



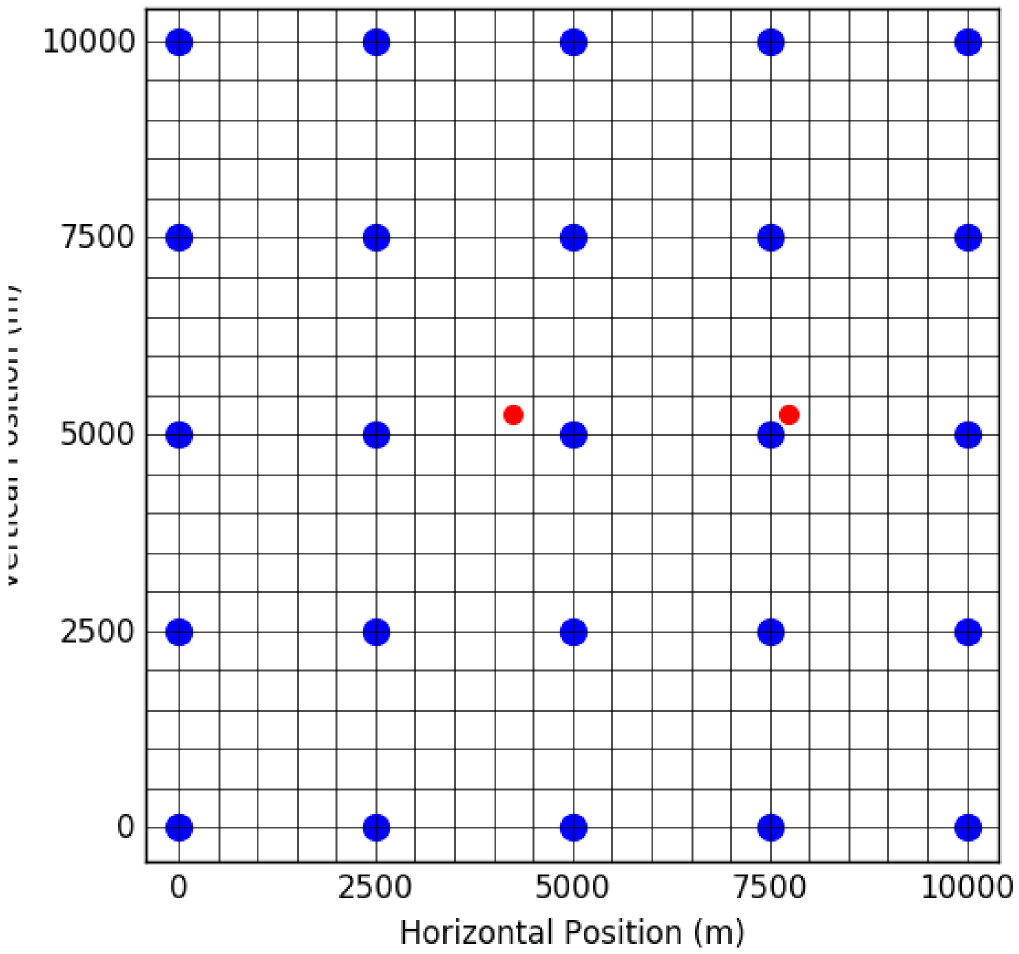
One Interference Source



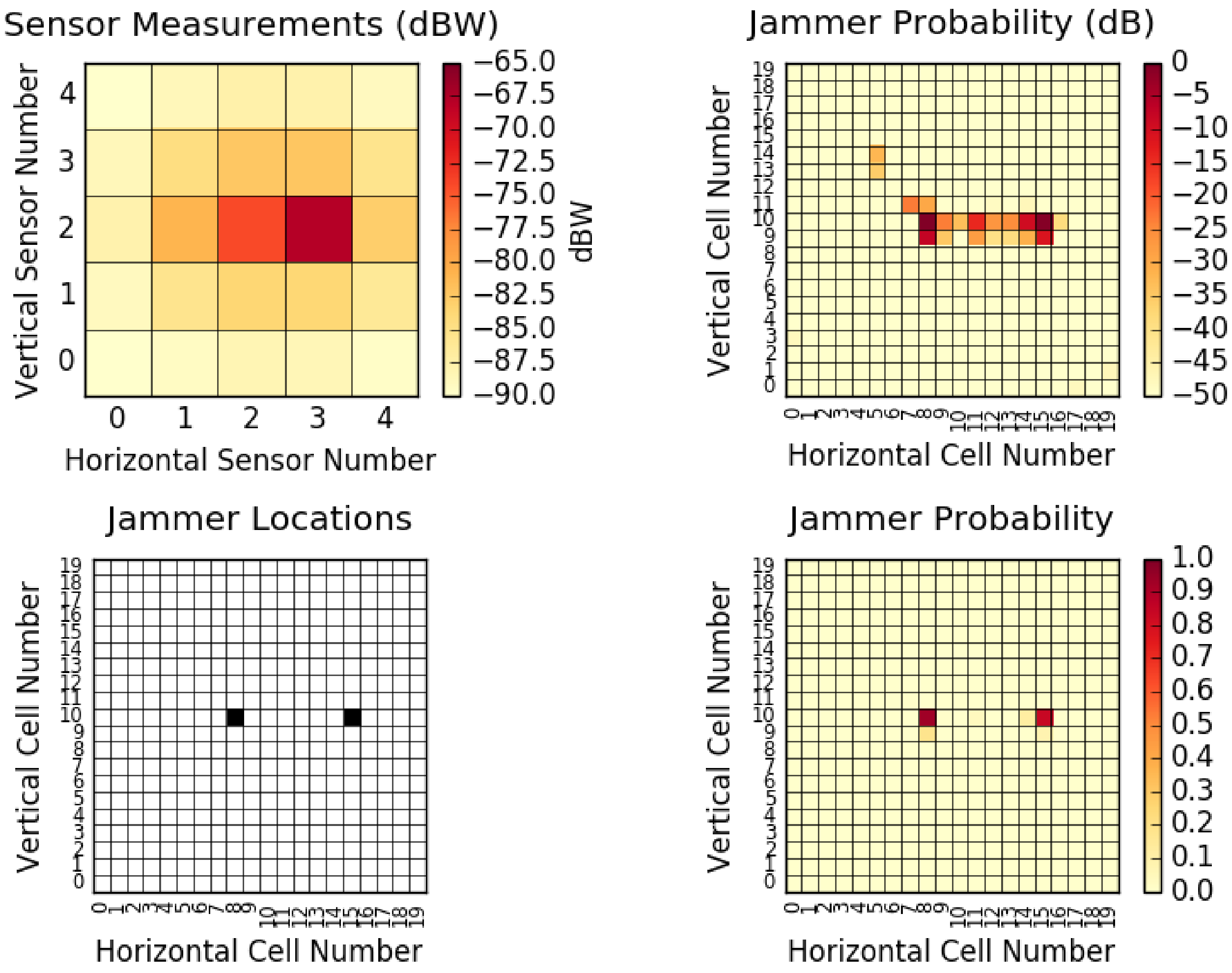
One Interference Source



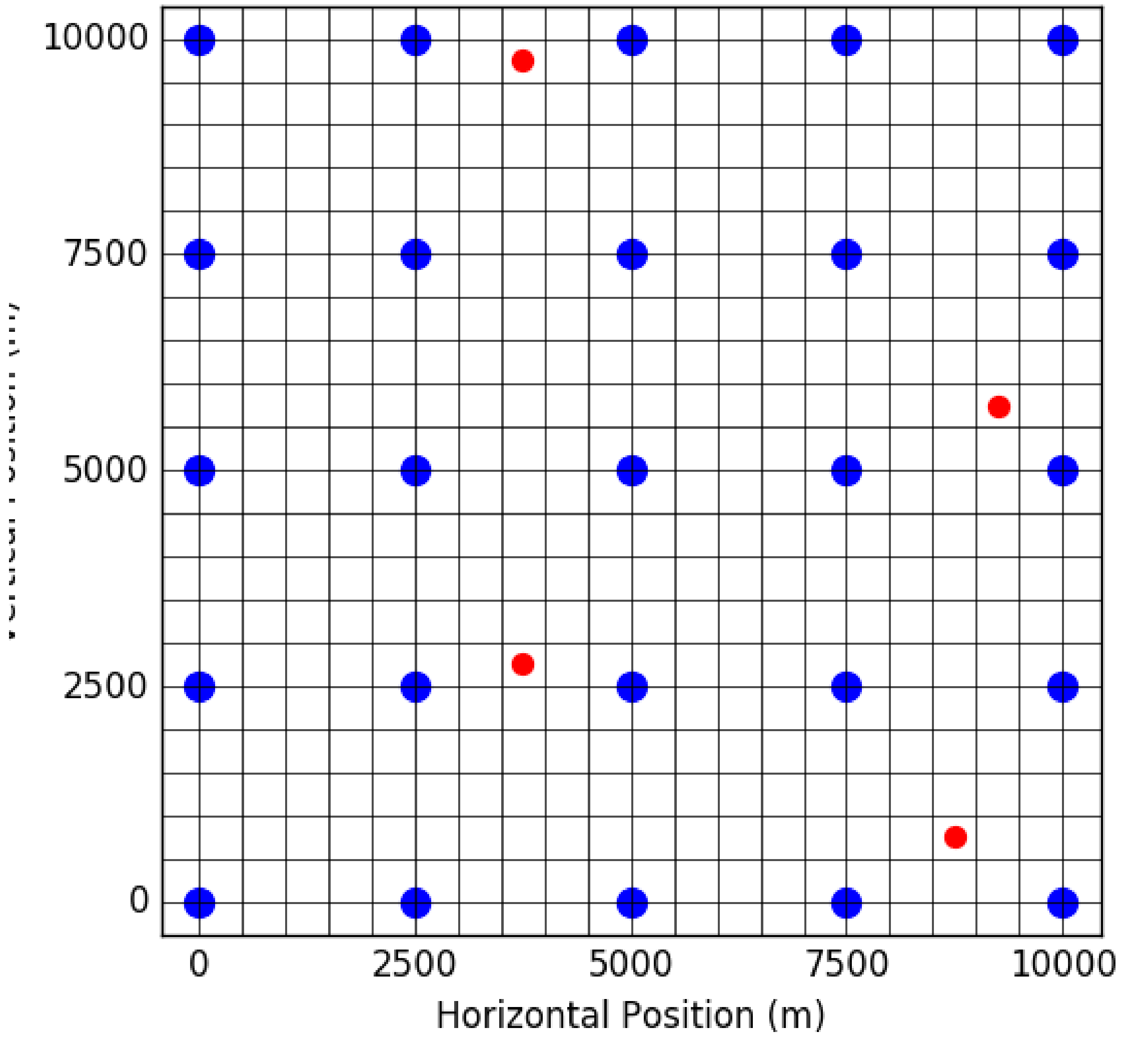
Two Interference Sources



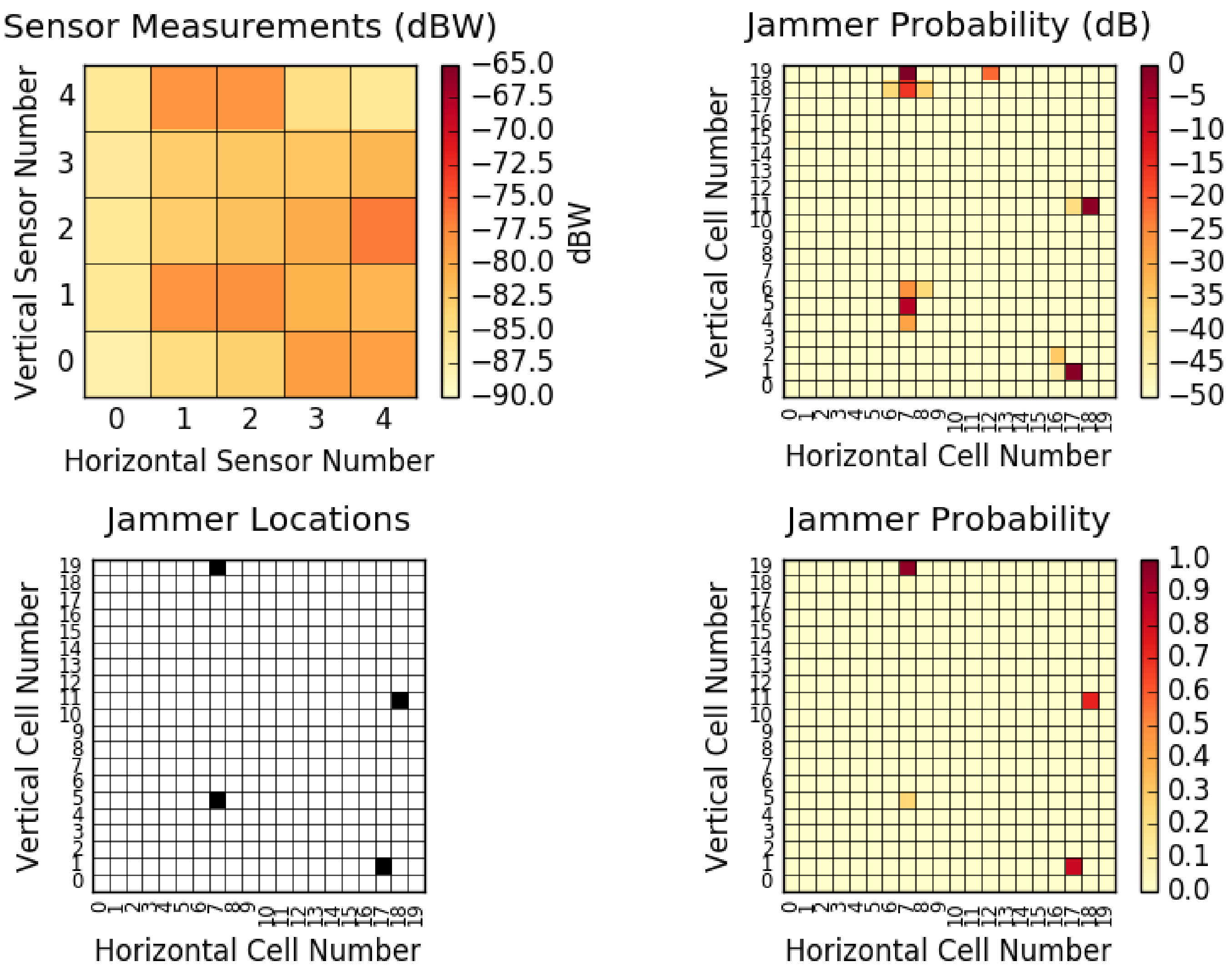
Two Interference Sources



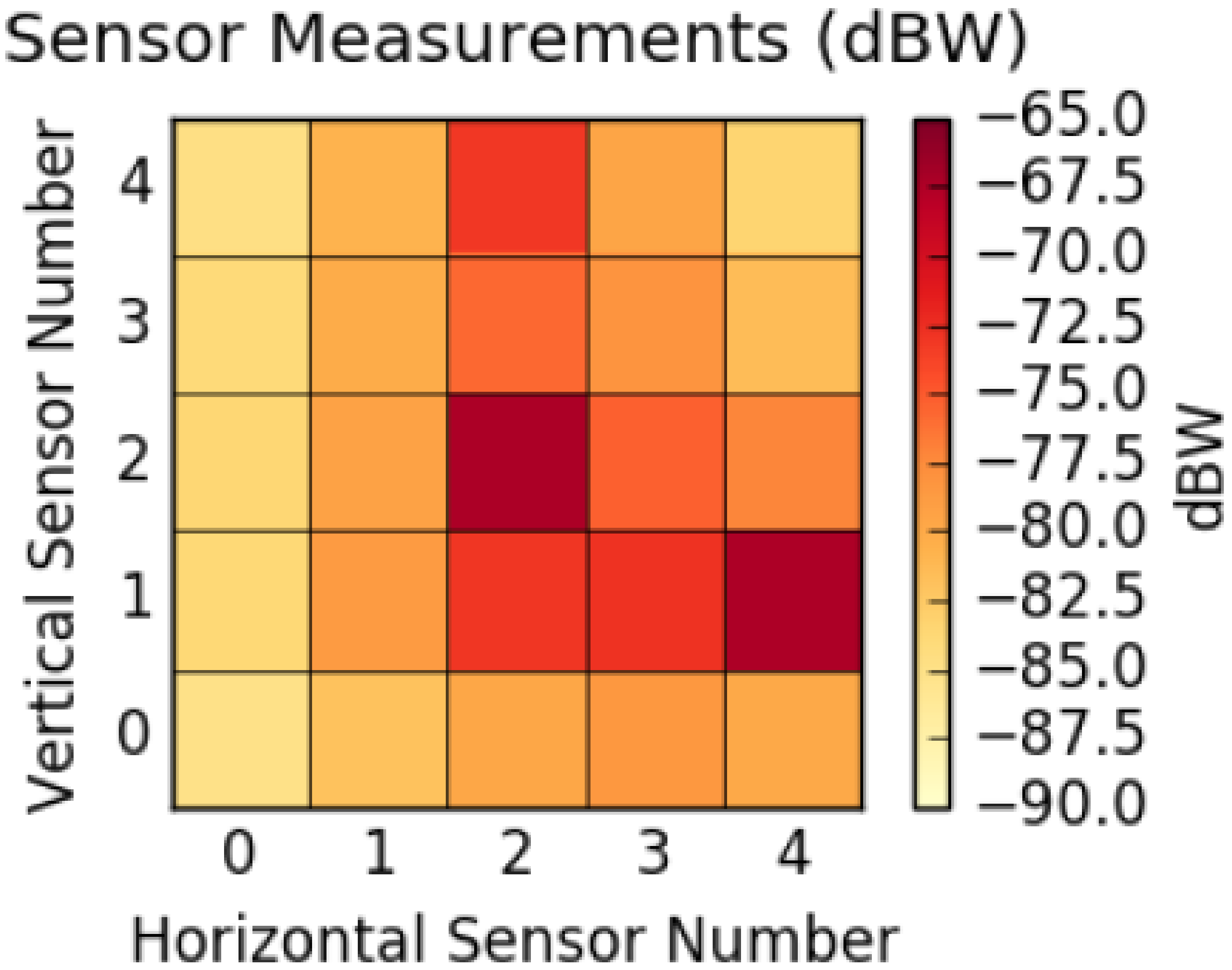
Four Interference Sources



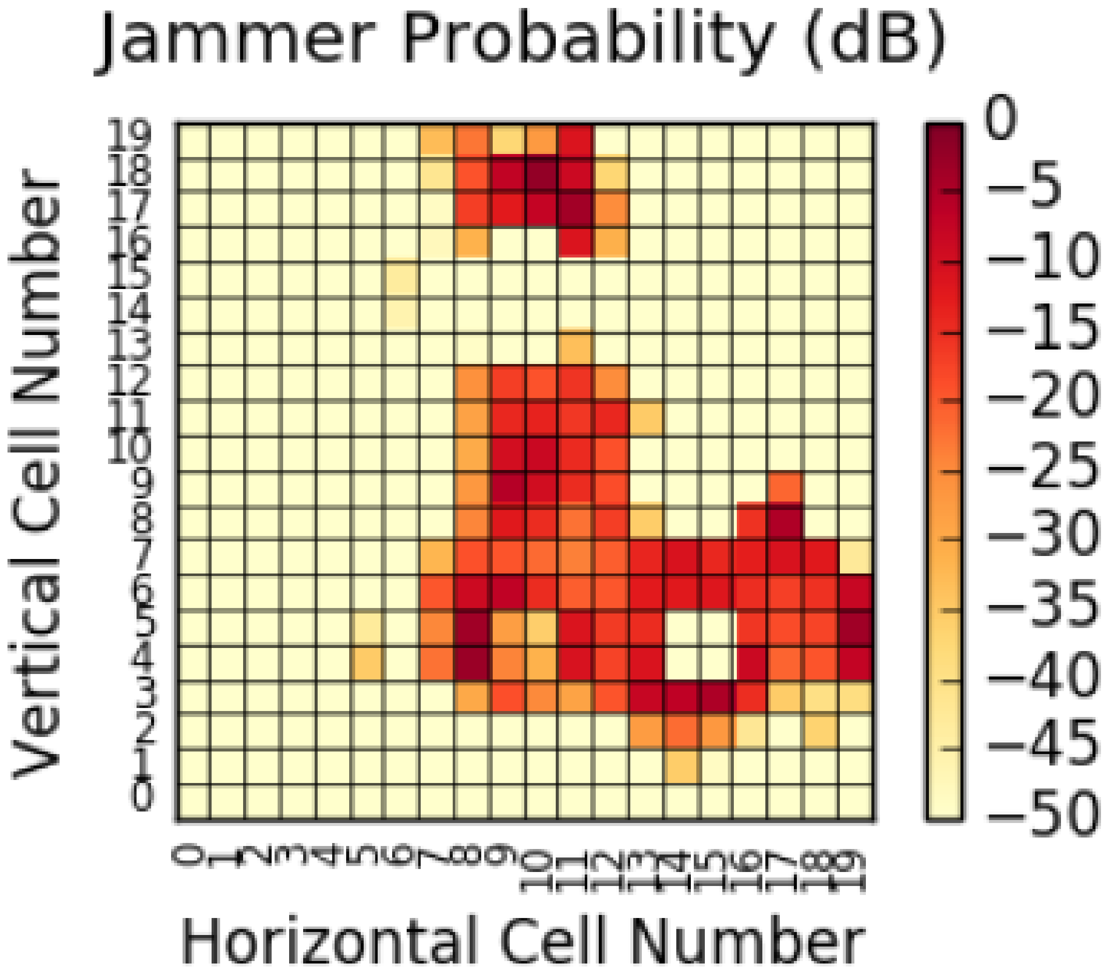
Four Interference Sources

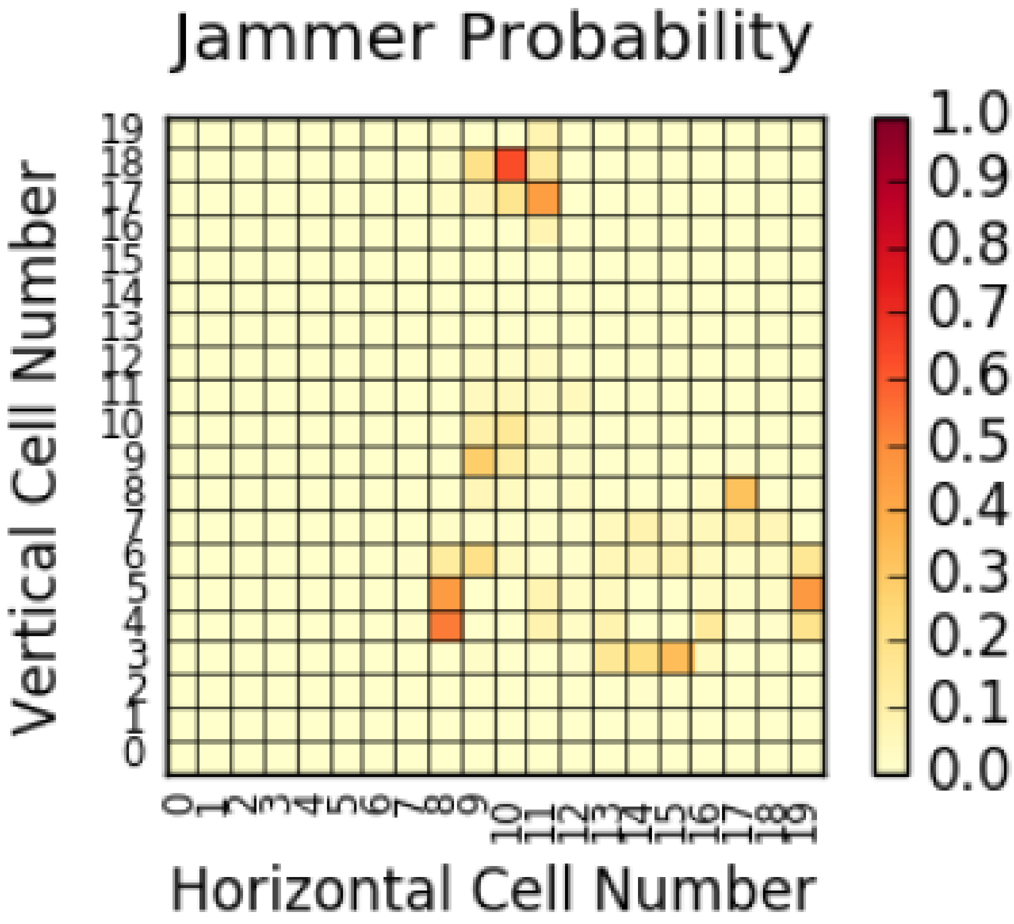


How many interference sources?



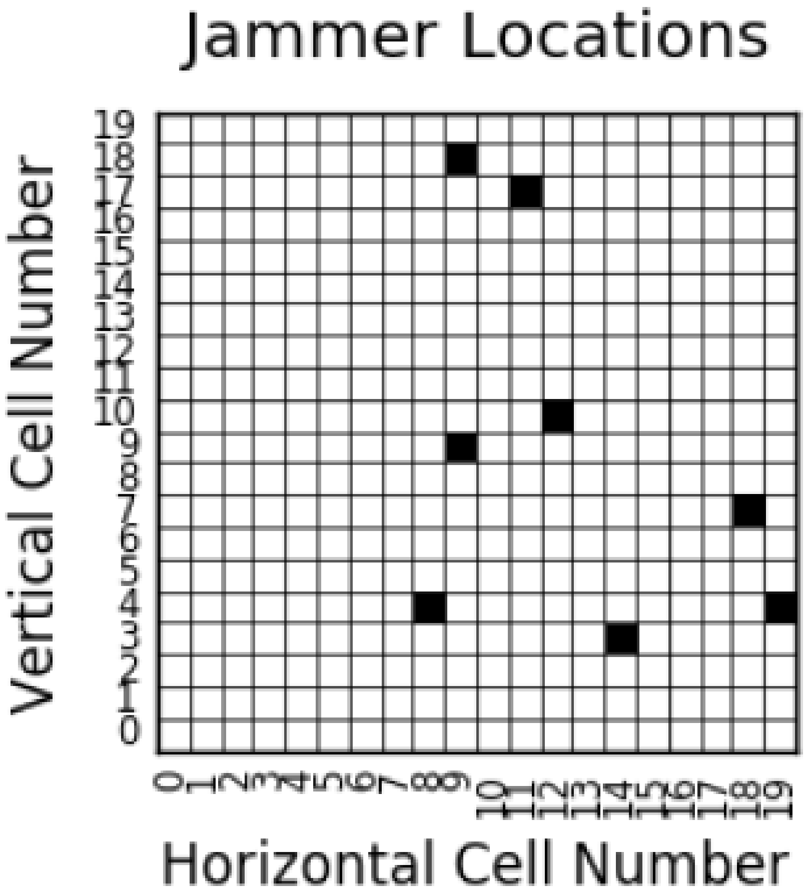
How many interference sources?

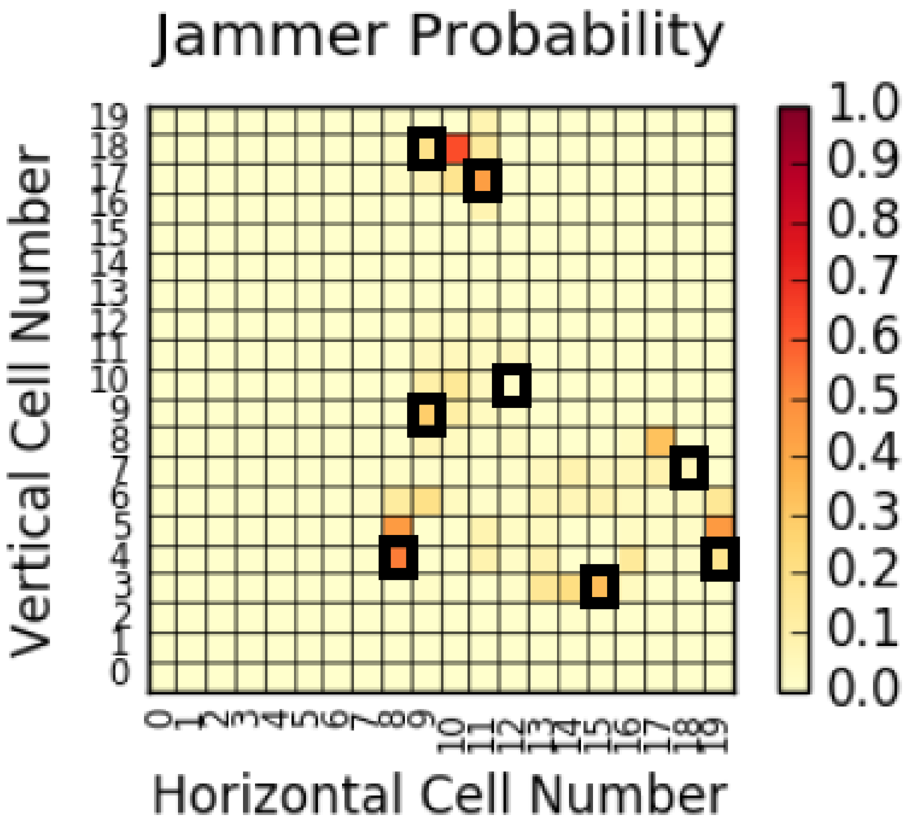




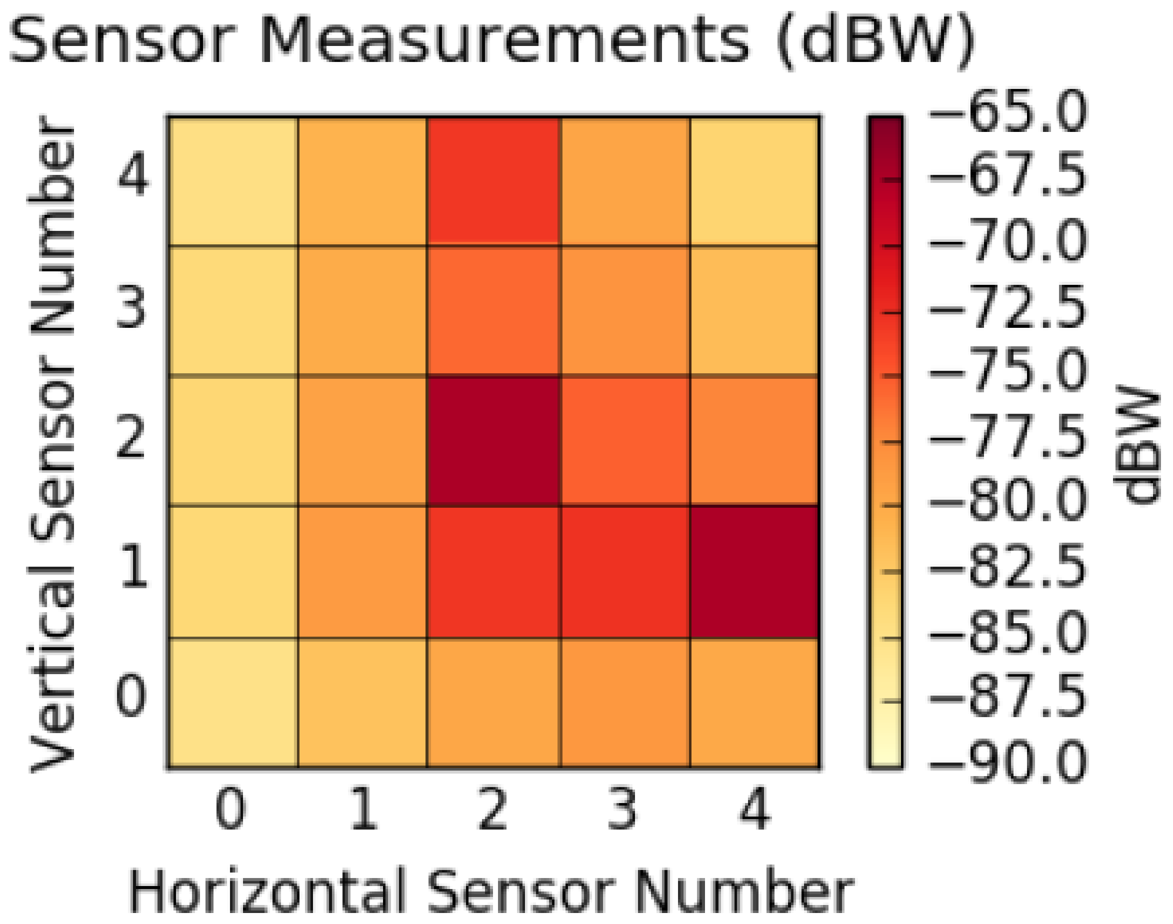
How many interference sources?

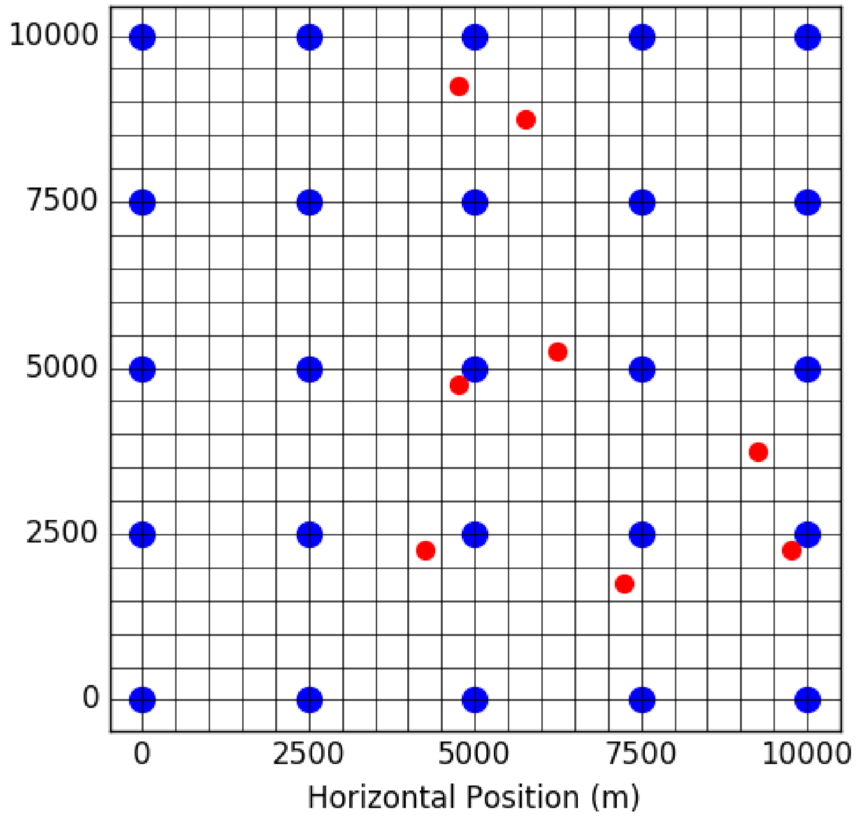
- 8 interference sources.



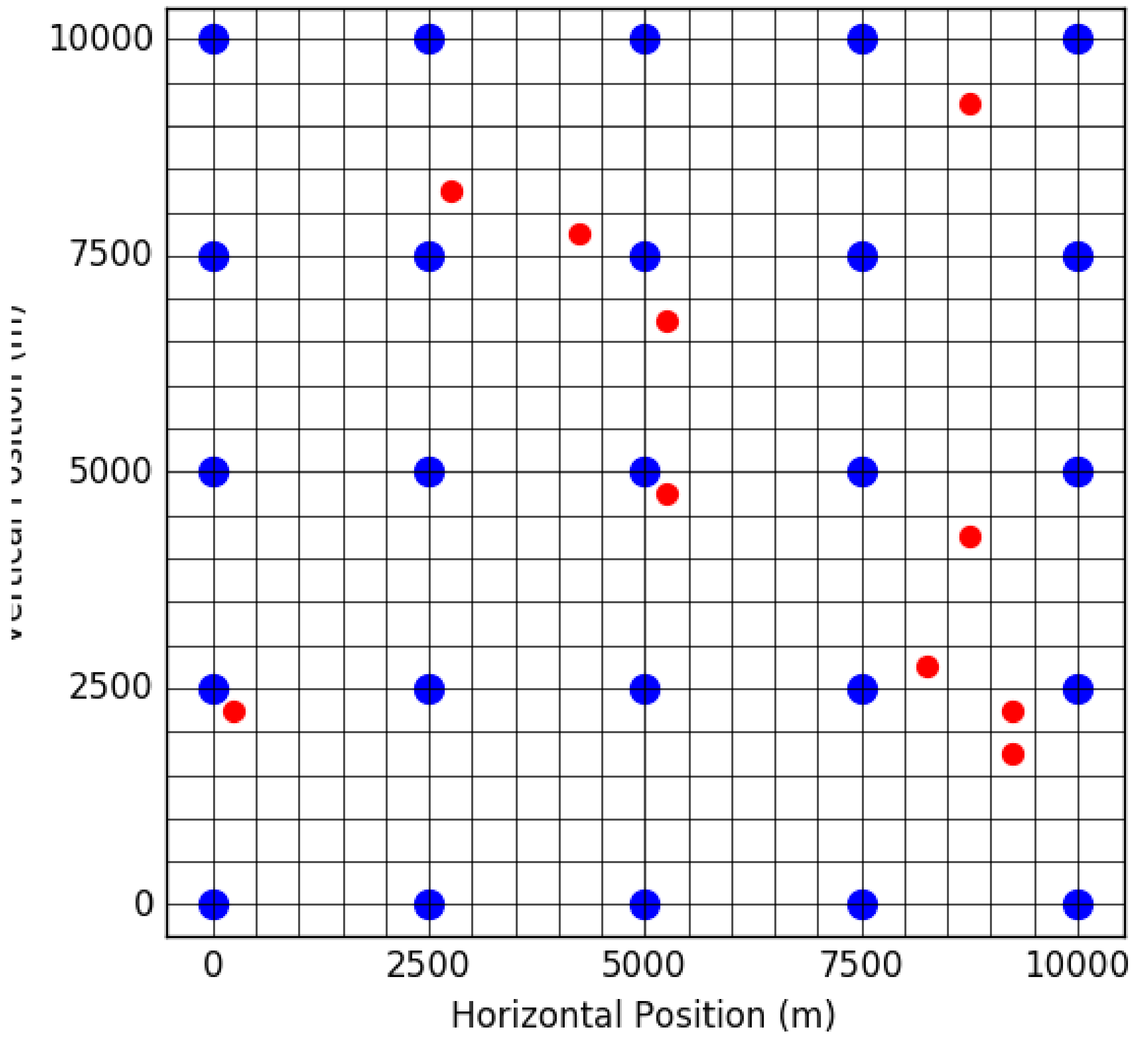


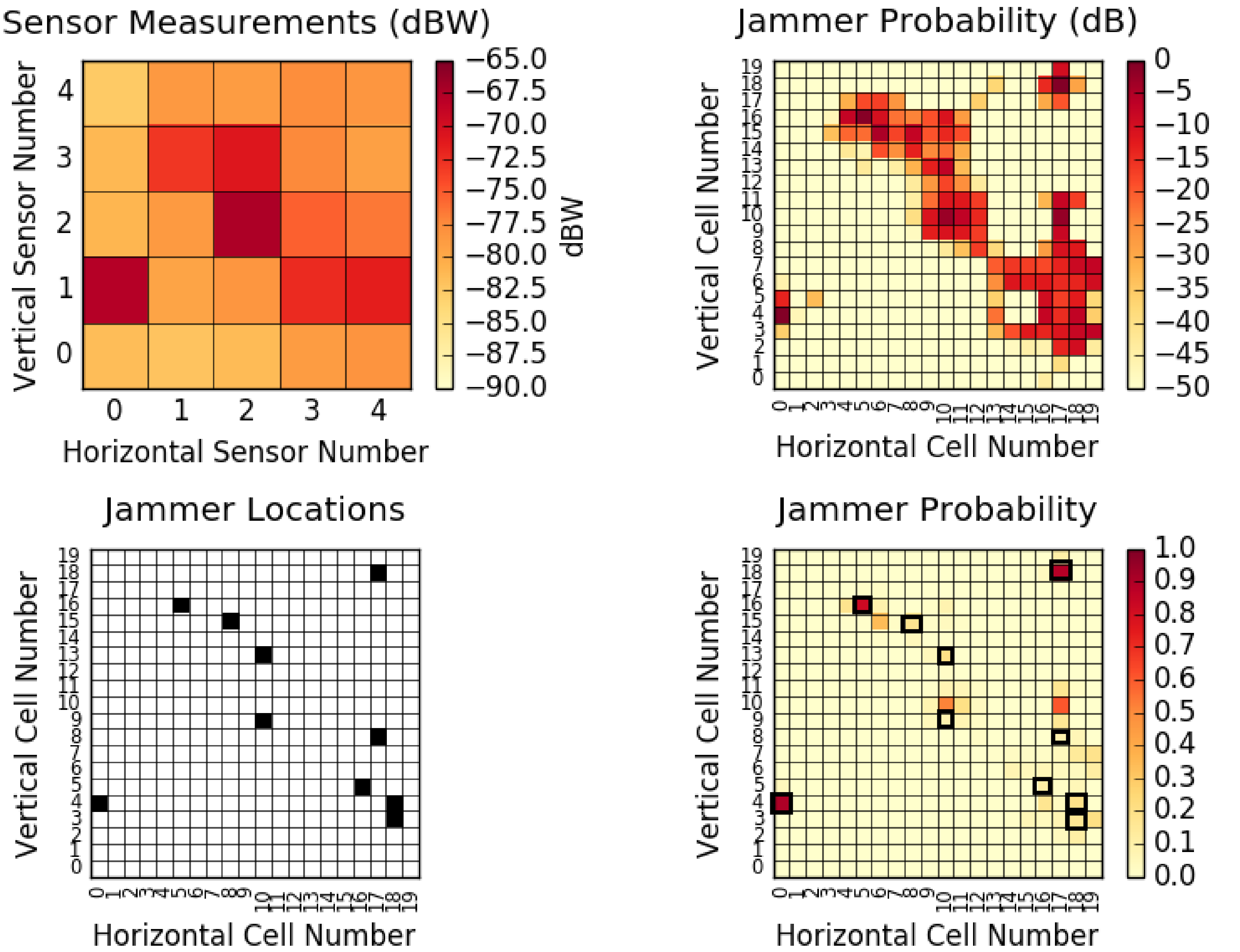
Eight Interference Sources





Ten Interference Sources

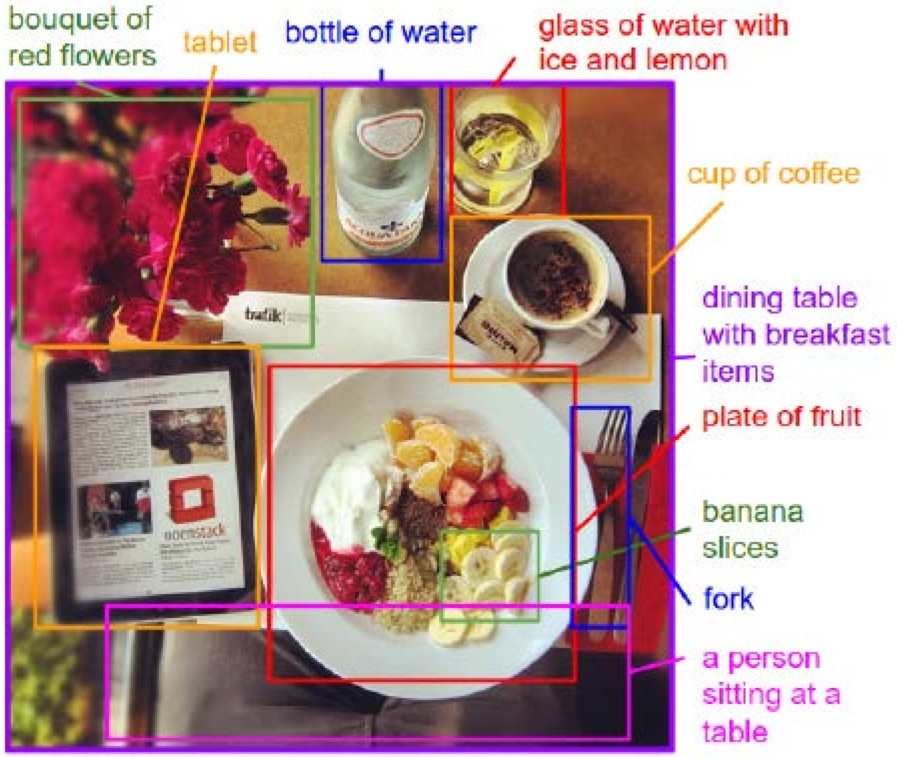



Sensor Measurements (dBW)
Vertical Sensor Number
Horizontal Sensor Number
dBW
Jammer Probability (dB)
Vertical Cell Number
Horizontal Cell Number
Jammer Locations
Vertical Cell Number
Horizontal Cell Number
Jammer Probability
Vertical Cell Number
Horizontal Cell Number

Conclusion and Future Work

- This framework has the potential to open up the interference source localization problem to a wide variety of machine learning techniques.

- Future Work:
    - Incorporate more sophisticated propagation models
    - Perform simulations on accurate terrain models
    - Tune the neural network macro parameters to improve performance
    - Implement technique on a testbed
    - Extend framework to utilize convolutional and recurrent neural networks so as to localize mobile jammers and non-uniform beam patterns.



"a young boy is holding a baseball bat."

http://cs.stanford.edu/people/karpathy/deepimagesent/

References

[1] A. A. Hussein, T. A. Rahman, and C. Y. Leow, "A survey and open issues of jammer localization techniques in wireless sensor networks," *Journal of Theoretical & Applied Information Technology*, vol. 71, no. 2, 2015.

[2] M. Liu, N. Xu, and H. Li, "Multi-sensor multi-target passive locating and tracking," *International Journal of Control, Automation, and Systems*, vol. 5, no. 2, pp. 200–207, 2007.

[3] T. Cheng, P. Li, and S. Zhu, "Multi-jammer localization in wireless sensor networks," in *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pp. 736–740, IEEE, 2011.

[4] K. Gromov, D. Akos, S. Pullen, P. Enge, and B. Parkinson, "Gidl: generalized interference detection and localization system," *ION GPS, Salt Lake City, UT*, vol. 20, no. 0, p. 0, 2000.

[5] J. Lindström, D. Akos, O. Isoz, and M. Junered, "Gnss interference detection and localization using a network of low cost front-end modules," in *International Technical Meeting of the Satellite Division of the Institute of Navigation: 24/09/2007-28/09/2007*, pp. 1165–1172, Institute of Navigation, The, 2007.

[6] M. Trinkle and D. Gray, "Interference localisation trials using adaptive antenna arrays," in *ION GPS*, vol. 2002, p. 15th, 2002.

[7] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–34, 1996.

[8] A. Ng, "Machine learning." Online. https://www.coursera.org/learn/machine-learning.

[9] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] F. A. Administration, "Faa doubles 'blanket' altitude for many uas flights." Online, 3 2016. https://www.faa.gov/news/updates/?newsId=85264.

[11] scikit-learn developers, "Neural network models (supervised)." Online, 7 2017. http://scikit-learn.org/stable/modules/neural_networks_supervised.html.