Hierarchical Temporal Memory for Air and Missile Defense Applications

Ezra Aylaian

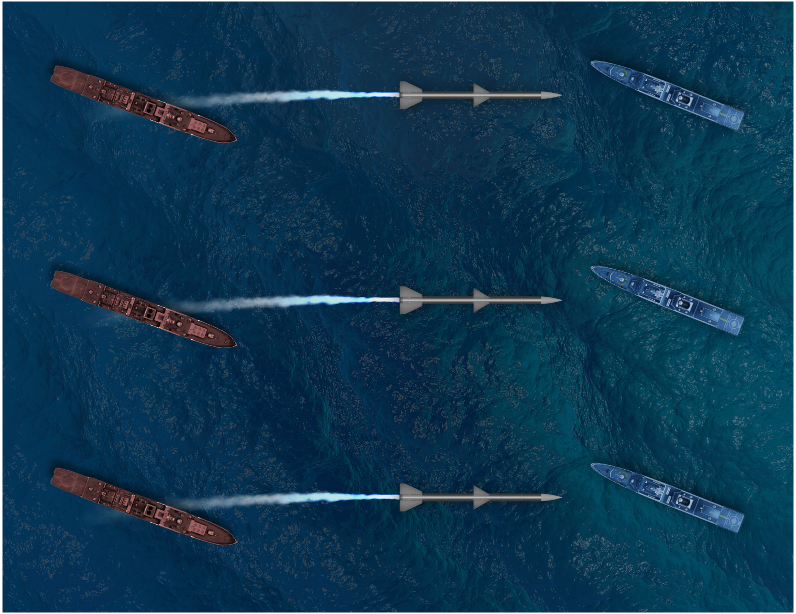
Maritime Force Engagement Control Group
Air and Missile Defense Sector
Johns Hopkins University Applied Physics Laboratory

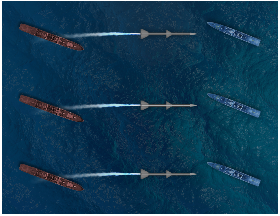
Department of Mathematics
University of Maryland

Joint work with Thomas Corcoran, Samim Manizade, and Jessica Stietzel

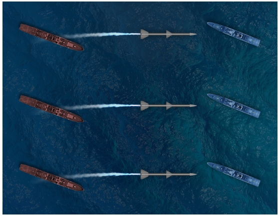
MORS Emerging Techniques Forum
December 5th, 2023

Table of Contents



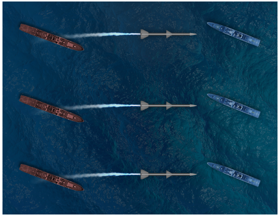


Raid 1: Benign

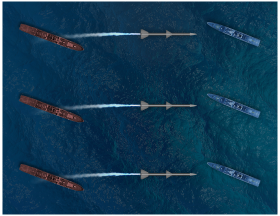


Raid 1: Benign

⋮

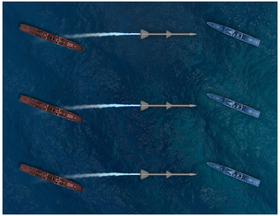


Raid 6: Benign

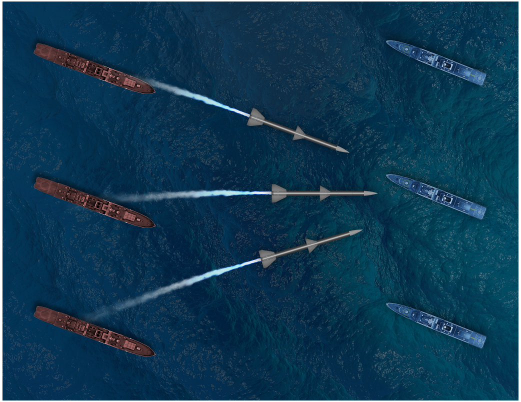


Raid 1: Benign

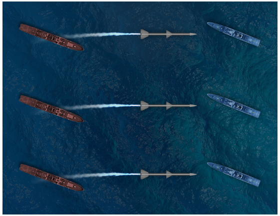
⋮



Raid 6: Benign

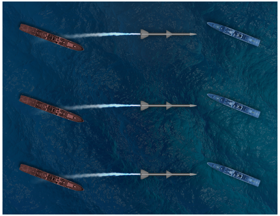


Raid 7: Anomalous

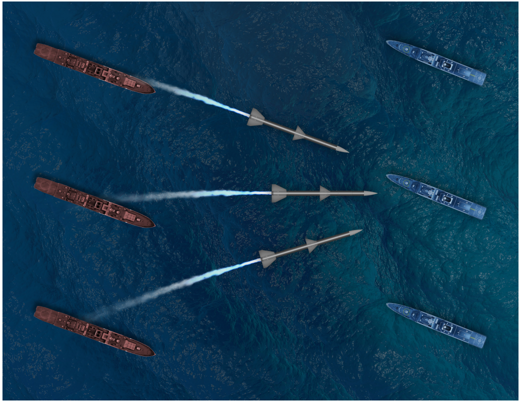


Raid 1: Benign

⋮



Raid 6: Benign



Raid 7: Anomalous

How can we detect a sudden change in tactics?

Table of Contents

Two-Element Boolean Algebra

Definition: The Two-Element Boolean Algebra

The **two-element boolean algebra** *B* is the set $\{0, 1\}$ endowed with the operations $\wedge$, $\vee$ and $\neg$ (and, or, not). 0 and 1 are interpreted as false and true.

For example, $0 \wedge 1 = 0$, and $0 \vee 1 = 1$, and $\neg 0 = 1$.

Two-Element Boolean Algebra

Definition: The Two-Element Boolean Algebra

The **two-element boolean algebra** B is the set $\{0, 1\}$ endowed with the operations $\wedge$, $\vee$ and $\neg$ (and, or, not). 0 and 1 are interpreted as false and true.

For example, $0 \wedge 1 = 0$, and $0 \vee 1 = 1$, and $\neg 0 = 1$. The following laws hold:

- $\wedge$ and $\vee$ are associative, commutative, and distribute over each other
- $a \vee (a \wedge b) = a$ and $a \wedge (a \vee b) = a$
- $a \vee 0 = a$ and $a \wedge 1 = a$
- $a \vee \neg a = 1$ and $a \wedge \neg a = 0$

The Boolean Algebra B^n

Definition: The Boolean Algebra B^n

B^n is the set $\{(a_1, \ldots, a_n) \mid a_i \in B\}$ endowed with the operations $\wedge$, $\vee$, and $\neg$ defined componentwise:

- $(a_1, \ldots, a_n) \wedge (b_1, \ldots, b_n) = (a_1 \wedge b_1, \ldots, a_n \wedge b_n)$
- $(a_1, \ldots, a_n) \vee (b_1, \ldots, b_n) = (a_1 \vee b_1, \ldots, a_n \vee b_n)$
- $\neg(a_1, \ldots, a_n) = (\neg a_1, \ldots, \neg a_n)$

For example, in B^3, we have $(0, 1, 0) \wedge (1, 1, 0) = (0, 1, 0)$, and $(0, 1, 0) \vee (1, 1, 0) = (1, 1, 0)$, and $\neg(0, 1, 0) = (1, 0, 1)$.

The Boolean Algebra B^n

Definition: The Boolean Algebra B^n

B^n is the set $\{(a_1, \ldots, a_n) \mid a_i \in B\}$ endowed with the operations $\wedge$, $\vee$, and $\neg$ defined componentwise:

- $(a_1, \ldots, a_n) \wedge (b_1, \ldots, b_n) = (a_1 \wedge b_1, \ldots, a_n \wedge b_n)$
- $(a_1, \ldots, a_n) \vee (b_1, \ldots, b_n) = (a_1 \vee b_1, \ldots, a_n \vee b_n)$
- $\neg(a_1, \ldots, a_n) = (\neg a_1, \ldots, \neg a_n)$

For example, in B^3, we have $(0,1,0) \wedge (1,1,0) = (0,1,0)$, and $(0,1,0) \vee (1,1,0) = (1,1,0)$, and $\neg(0,1,0) = (1,0,1)$. All of the laws from the previous slide hold for B^n too:

- $\wedge$ and $\vee$ are associative, commutative, and distribute over each other
- $a \vee (a \wedge b) = a$ and $a \wedge (a \vee b) = a$
- $a \vee (0, \ldots, 0) = a$ and $a \wedge (1, \ldots, 1) = a$
- $a \vee \neg a = (1, \ldots, 1)$ and $a \wedge \neg a = (0, \ldots, 0)$

Sparse Distributed Representations

Elements of B^n can be classified by how many of their components have ones. Let B^n_w be elements of B^n with w ones, then $B^n = \cup_{w=0}^{n} B^n_w$.

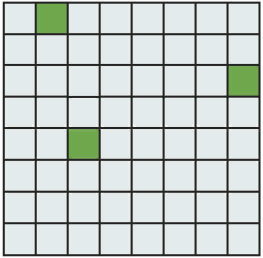
Sparse Distributed Representations

Elements of B^n can be classified by how many of their components have ones. Let B^n_w be elements of B^n with w ones, then $B^n = \cup_{w=0}^n B^n_w$. If $w \ll n$, then elements of B^n_w are called **Sparse Distributed Representations (SDRs)** of size n and sparsity w/n.

Sparse Distributed Representations

Elements of B^n can be classified by how many of their components have ones. Let B^n_w be elements of B^n with w ones, then $B^n = \cup^n_{w=0} B^n_w$. If $w \ll n$, then elements of B^n_w are called **Sparse Distributed Representations (SDRs)** of size n and sparsity w/n.

SDRs are often denoted using arrays:



Hierarchical Temporal Memory

- **Hierarchical Temporal Memory (HTM)** is a biologically-inspired machine learning algorithm created by Numenta based on a series of conjectures about the structure of the neocortex (the Thousand Brains Theory).

Hierarchical Temporal Memory

- **Hierarchical Temporal Memory (HTM)** is a biologically-inspired machine learning algorithm created by Numenta based on a series of conjectures about the structure of the neocortex (the Thousand Brains Theory).
- HTM features online learning, which means that it learns the scenario and generates results in real time with no prior training.

Hierarchical Temporal Memory

- **Hierarchical Temporal Memory (HTM)** is a biologically-inspired machine learning algorithm created by Numenta based on a series of conjectures about the structure of the neocortex (the Thousand Brains Theory).
- HTM features online learning, which means that it learns the scenario and generates results in real time with no prior training.

- The **encoder** maps the input space into the space of SDRs in a way that preserves semantic structure. The **Spatial Pooler (SP)** learns to represent the outputs of the encoder at a fixed low sparsity while again preserving semantic structure. The **Temporal Memory (TM)** learns to predict which components will be ones in the next SP output given the previous SP outputs and gives an **anomaly score** based on how inaccurate it was. Finally, the anomaly score is converted into an **anomaly likelihood**, the probability that an anomaly occurred.

Applications

We focus on HTM's anomaly detection. An **anomaly** is a deviation from the historic norm.

Applications

We focus on HTM's anomaly detection. An **anomaly** is a deviation from the historic norm. Online anomaly detection has many applications:

1. Detect anomalous behavior of individual white or red naval assets.

Applications

We focus on HTM's anomaly detection. An **anomaly** is a deviation from the historic norm. Online anomaly detection has many applications:

1. Detect anomalous behavior of individual white or red naval assets.
2. Mid-battle change of tactics detection (battle level).

Applications

We focus on HTM's anomaly detection. An **anomaly** is a deviation from the historic norm. Online anomaly detection has many applications:

1. Detect anomalous behavior of individual white or red naval assets.
2. Mid-battle change of tactics detection (battle level).
3. Change of phase of flight detection for incoming airborne threats.

Applications

We focus on HTM's anomaly detection. An **anomaly** is a deviation from the historic norm. Online anomaly detection has many applications:

1. Detect anomalous behavior of individual white or red naval assets.
2. Mid-battle change of tactics detection (battle level).
3. Change of phase of flight detection for incoming airborne threats.

Often we want to detect *threats*, but HTM detects *anomalies*! E.g. we want to do wolf-in-sheep's-clothing detection of disguised red assets listed as white, but the best anomaly detection can do is say whether the behavior of an asset listed as white is deviating from its historic norm.

Going from anomaly detection to threat detection can be difficult.

Table of Contents

Methodology

How useful is HTM for air and missile defense?

Methodology

How useful is HTM for air and missile defense?

Properties that we want HTM to have:

1. **Noise resilience:** the ability to operate effectively on noisy data.

Methodology

How useful is HTM for air and missile defense?

Properties that we want HTM to have:

1. **Noise resilience:** the ability to operate effectively on noisy data.
2. **Selective attention:** the ability to operate effectively on an input space where only some data is relevant.

Methodology

How useful is HTM for air and missile defense?

Properties that we want HTM to have:

1. **Noise resilience:** the ability to operate effectively on noisy data.
2. **Selective attention:** the ability to operate effectively on an input space where only some data is relevant.
3. **Studiousness:** the utility of the paradigmal particularities of the learning algorithms.

Methodology

How useful is HTM for air and missile defense?

Properties that we want HTM to have:

1. **Noise resilience:** the ability to operate effectively on noisy data.
2. **Selective attention:** the ability to operate effectively on an input space where only some data is relevant.
3. **Studiousness:** the utility of the paradigmal particularities of the learning algorithms.
4. **Ease-of-use and technical readiness** the ability to be easily operationalized.

Methodology

How useful is HTM for air and missile defense?

Properties that we want HTM to have:

1. **Noise resilience:** the ability to operate effectively on noisy data.
2. **Selective attention:** the ability to operate effectively on an input space where only some data is relevant.
3. **Studiousness:** the utility of the paradigmal particularities of the learning algorithms.
4. **Ease-of-use and technical readiness** the ability to be easily operationalized.

We also run two experiments designed to evaluate the experimental plausibility of HTM at detecting anomalies in naval asset behavior and in missile raid behavior.

Noise Resilience

The ability to operate effectively on noisy data

To test the noise resilience of HTM's spatial pooler, we used well-known MNIST digit classification problem, which contains 60,000 handwritten grayscale 28x28 images of digits for training and 10,000 for testing [4]. Noise was added to the digits as follows:

```
# image is a numpy array, noise_level is an integer
def add_noise(image, noise_level):
      noise = np.rint(np.random.normal(scale=noise_level, size=(28,28)))
      return np.clip(np.absolute(image + noise), 0, 255)
```

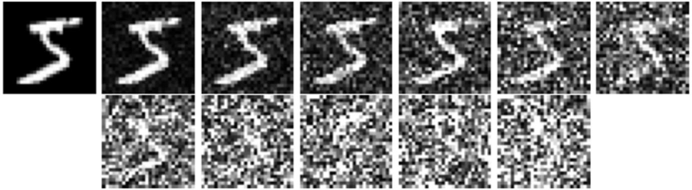
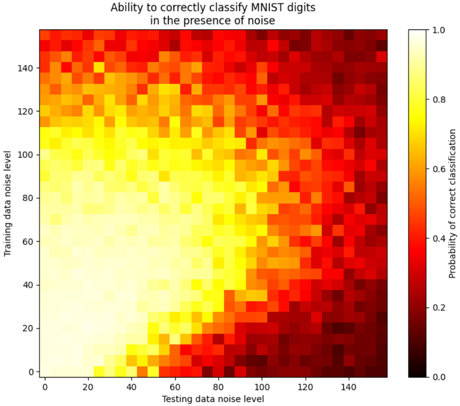
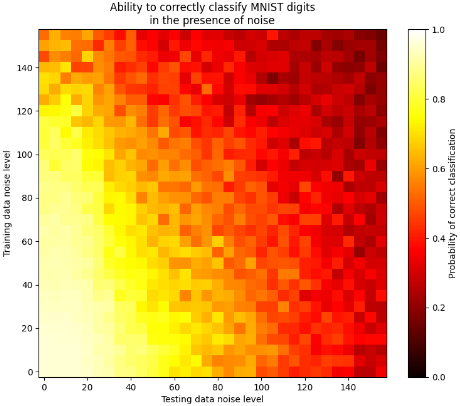


Figure 1: A handwritten MNIST digit with noise levels 0, 20, 40, 60, ..., 220.





CNN, 1 epoch, 2 Monte Carlos

HTM, 1 epoch, 2 Monte Carlos

Convolutional neural network (CNN) is a well-known algorithm that we use as a benchmark.
Monte Carlo is sometimes abbreviated as MC.

Selective Attention

The ability to operate effectively on an input space where only some data is relevant

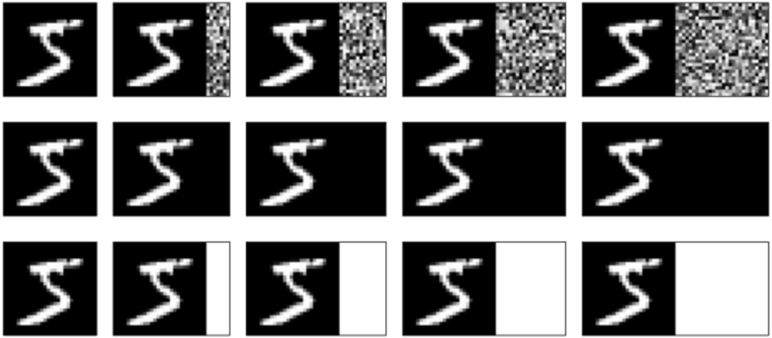
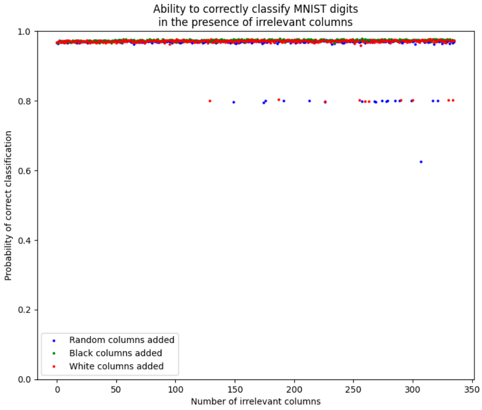
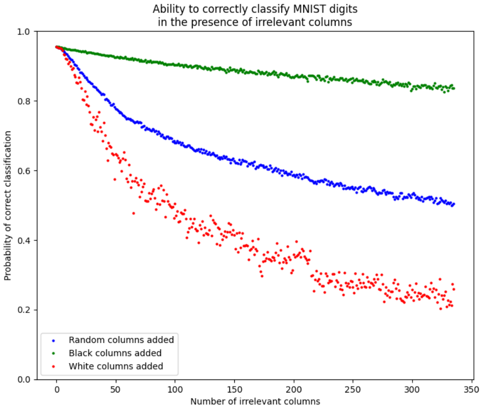


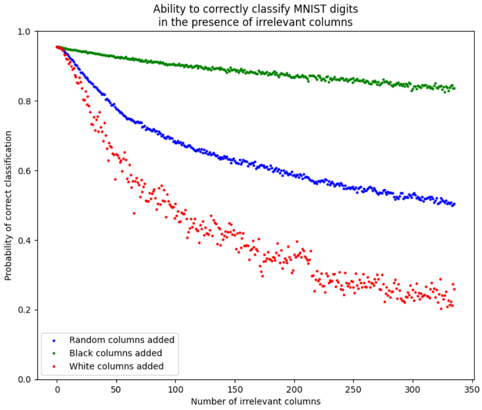
Figure 2: A handwritten MNIST digit with 0, 7, 14, 21, and 28 noisy, black, and white irrelevant columns added.



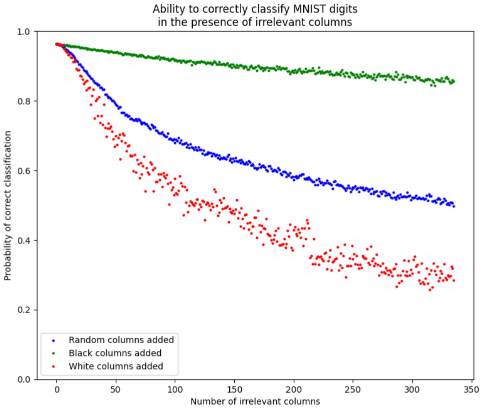
CNN, 1 epoch, 5 Monte Carlos



HTM, 1 epoch, 5 Monte Carlos



HTM, 1 epoch, 5 Monte Carlos

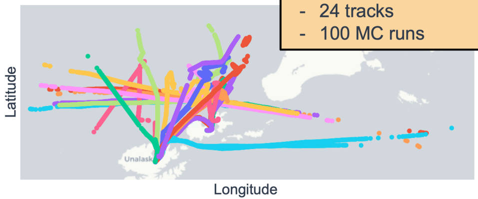


HTM, 3 epochs, 3 Monte Carlos

Table of Contents

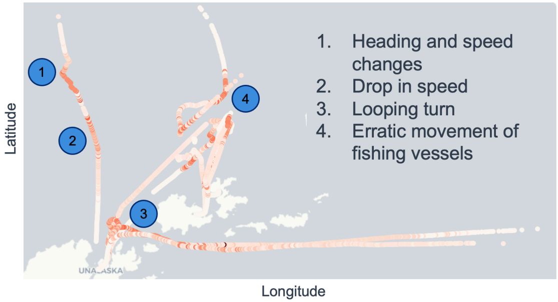
Real-world Automatic Identification System (AIS) track data of ships off the coast of Alaska. We analyze each track separately to detect which have anomalies.

STARE input AIS tracks
Colored by AIS track ID



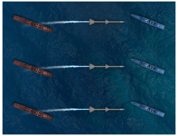
Alaska Example Scenario
- 48 hours of data from January 2017
- 24 tracks
- 100 MC runs

AIS flagged tracks
Colored by Anomaly Log Likelihood



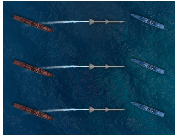
At each timestep, the track's position (latitude/longitude), heading, and speed are encoded as SDRs. HTM analyzes the sequence of SDRs for anomalies.

Table of Contents

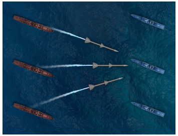


Raid 1: Benign

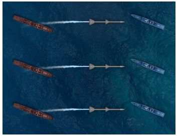
⋮



Raid 6: Benign

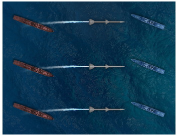


Raid 7: Anomalous

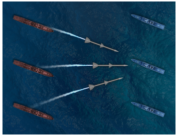


Raid 1: Benign

⋮

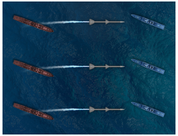


Raid 6: Benign



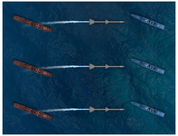
Raid 7: Anomalous

- Each missile's position (lat/lon) is encoded as an SDR.

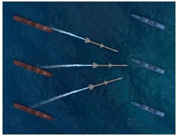


Raid 1: Benign

⋮

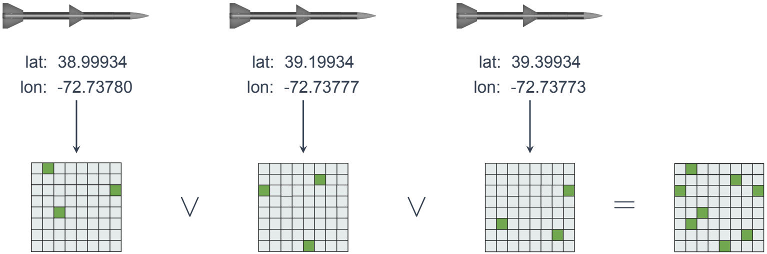


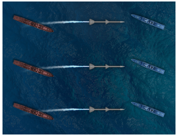
Raid 6: Benign



Raid 7: Anomalous

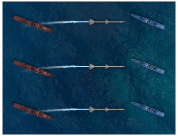
- Each missile's position (lat/lon) is encoded as an SDR.
- The missile SDRs are $\vee$ed together to get the scenario SDR.



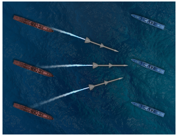


Raid 1: Benign

⋮

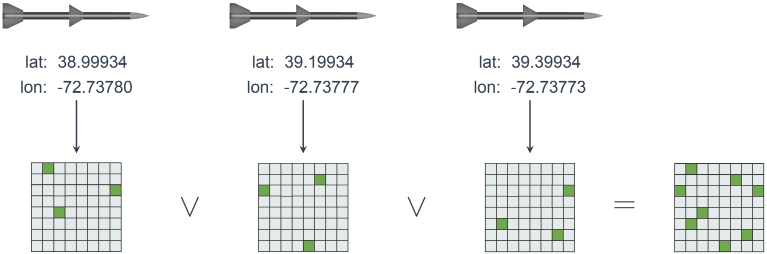


Raid 6: Benign

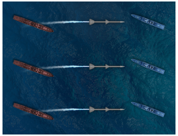


Raid 7: Anomalous

- Each missile’s position (lat/lon) is encoded as an SDR.
- The missile SDRs are $\vee$ed together to get the scenario SDR.

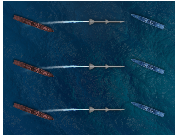


- We run the scenario SDRs through HTM to get an anomaly probability associated with each timestep.

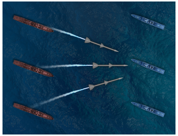


Raid 1: Benign

⋮

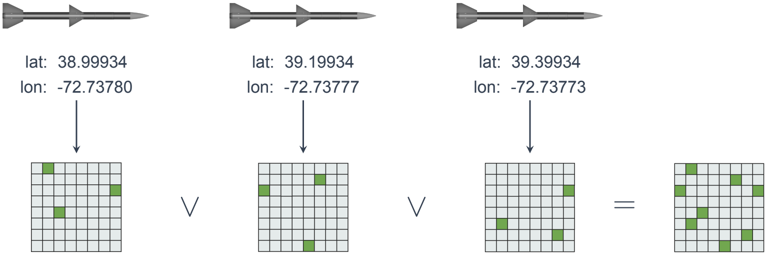


Raid 6: Benign



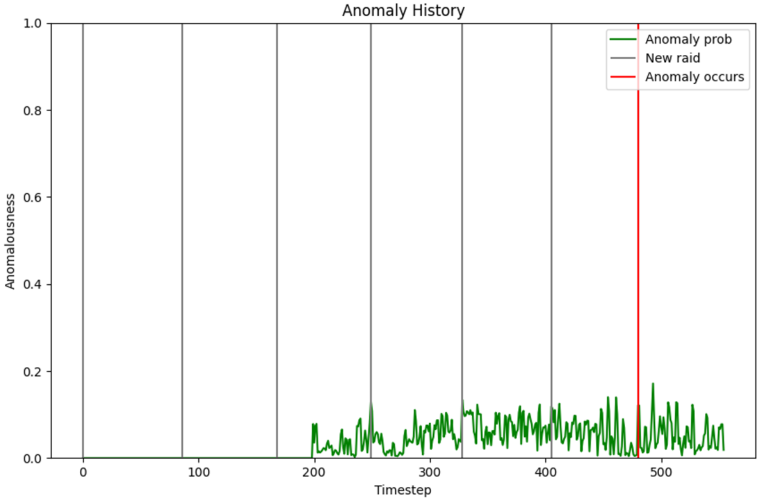
Raid 7: Anomalous

- Each missile's position (lat/lon) is encoded as an SDR.
- The missile SDRs are $\vee$ed together to get the scenario SDR.

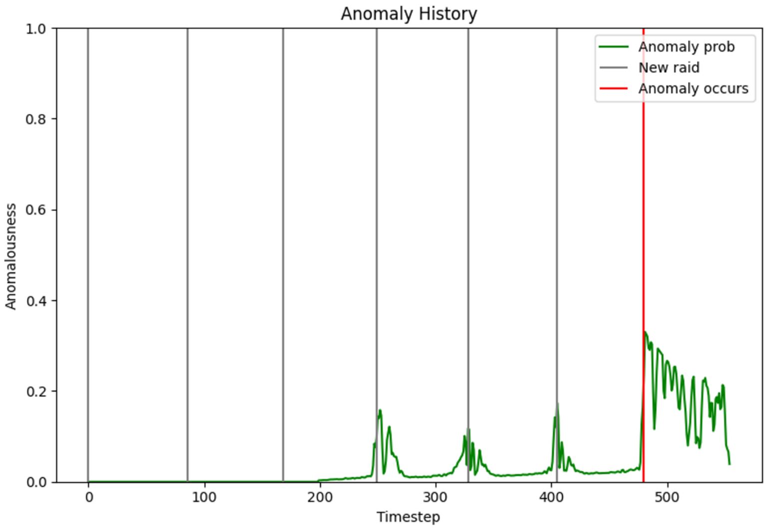


- We run the scenario SDRs through HTM to get an anomaly probability associated with each timestep.
- Success is achieved if and only if the anomaly probability spikes when the anomalous raid begins.

Results Without Hyperparameter Optimization



Results With Hyperparameter Optimization



Summary

Is HTM...

1. Noise resilient? Yes
2. Attention selective? Significantly worse than CNN
3. Studious? Yes, features online learning
4. Ready to operationalize? No

HTM performed well in two simple AMD scenarios.

JOHNS HOPKINS

APPLIED PHYSICS LABORATORY