Numenta

The Numenta Anomaly Benchmark

The first temporal benchmark designed for anomaly detection in streaming data

White Paper

Executive Summary

Across every industry, we are seeing an increase in the availability of streaming, time-series data. Largely driven by the rise of the Internet of Things (IoT) and connected real-time data sources, we now have an enormous number of applications with sensors that produce important data that changes over time[1]. But it is not desirable or practical to capture and store all of this information for analysis later. Businesses are quickly finding themselves drowning in databases full of rapidly decaying data that cannot be turned into actionable information. How can we draw valuable insights from this onslaught of streaming data?

One answer to this question is to figure out quickly when something is different in any given stream of data such that action can be taken right away. A change might be for a negative reason – the temperature sensor on an engine is going up, indicating a possible imminent failure – or the change might be for a positive reason – web clicks on a new product page are abnormally high, showing strong demand – but, either way, the change is interesting and may require action. We call such a change an anomaly, basically a data point or a series of data points other than what is expected.

Early anomaly detection in streaming data has practical and significant applications across many industries. Imagine getting early indications that:

- A bottleneck is developing in your IT infrastructure
- A vehicle in your logistics network is on an abnormal path
- Your web site is developing unusually high volume
- A medical sensor is indicating signs of patient distress

While it's easy to see why detecting real-time anomalies is important, it's very difficult to achieve. This paper explains why the problem is so hard and how the Numenta Anomaly Benchmark (NAB) can be used to assess the performance of various techniques. By using NAB, a business manager can compare his or her internal anomaly detection techniques to published algorithms. Numenta's hope is that a community will build around NAB to increase the number of labeled data files available for testing as well as the number of algorithms for comparison. NAB will enable business leaders to select the best anomaly detection algorithm for specific applications.

[1] We also refer to time-series data as temporal data.

The Challenge of Anomaly Detection in Streaming Data

It is surprisingly difficult to find anomalies in time series data. Most anomaly detection methods are designed for static, or spatial, data, meaning data that might have a correlation at one specific point in time, but does not take into account the sequence of data points over time. As a result, those methods are ineffective when applied to streaming data. The following example involving heart rate monitoring helps to illustrate the reasons why.

> Consider a hospital patient with a heart rate that is typically between 80 and 100 beats per minute. The patient is connected to a heart rate monitor, and the nurse wants to be notified when the heart rate falls out of the normal range. In this instance, the nurse might set a threshold, which is a common method for detecting anomalies. She might set a threshold at 75 on the low end and 105 on the high end such that an alarm will ring each time the monitor drops below 75 or raises above 105. This technique invites significant problems.
>
> First, different patients might have different normal heart rates. If the previous patient's normal was between 70 and 90, then the nurse would get false positives at 75, and would not be getting alerts she should have gotten at 95 (false negatives). Second, this method detects problems AFTER they occur, not before. By the time the heart hits the 105 trigger, it's already in trouble. Relatedly, the nurse will miss any strange behaviors that don't trigger the threshold. If the patient's heart beat suddenly starts jumping around wildly, but stays within the 75 and 105 boundaries, the alarm would not ring, but something might be terribly wrong.
>
> Another issue with this method is there is no learning. It is conceptually difficult to distinguish between an anomaly and a "new normal" that should be learned. Perhaps the patient whose old normal is 70 to 90 takes a new medication that raises her heart rate to a new normal of 90 to 110. At first, any anomaly detection technique should indicate anomalous behavior. But a good anomaly detection system should quickly learn the new normal and stop alarms at heart rates between 90 and 110. In the case of a simple threshold at 105, the nurse might receive many false positives until she manually resets the system to the new level. Alternatively, she might get so frustrated by the false alarms that she would turn them off or ignore them altogether.

How can we solve the problem of finding anomalies in streaming data if we don't use traditional methods like thresholds? Although different approaches exist to address this need, there is no benchmark to test and score these algorithms on streaming time-series data. We created the Numenta Anomaly Benchmark to fill this gap, i.e. to create a scoring mechanism that properly evaluates anomaly detection in time series data, such as giving "credit" for early detections and for learning new normal patterns.

The Numenta Anomaly Benchmark

The Numenta Anomaly Benchmark (NAB) is an open source framework designed to compare and evaluate algorithms for detecting anomalies in streaming data. Anomalies in streaming data are patterns that do not conform to past patterns of behavior for the given data stream.

There are two key components to NAB: the labeled dataset and the scoring system. Each was designed with the goal of creating the ideal evaluation framework for real-world anomaly detection algorithms.

The NAB Dataset

For this type of benchmark to be most effective and useful for the research community, it needs to contain real-world labeled data from multiple domains. This type of data is extremely valuable for performing comparisons and evaluations, yet very rare. Along with a few simulated data files, NAB incorporates real-world data with anomalies that have known causes as part of the benchmark. Numenta has accumulated these data files over several years of working with customers on solving their anomaly related problems. The NAB v1.0 dataset contains 58 data files, each with 1,000-22,000 data instances, for a total of 365,551 data points.

We set out three main requirements when we created the NAB dataset. First, we required a variety of streaming data anomaly types, e.g. having both true system failures as well as planned shutdowns. Different anomalies display different behavior, and it’s important to be able to test for as many as possible.

Second, we sought a variety of data metrics across multiple domains and applications. For example, a one-second delay in heart monitoring data could be significant, but the same delay in GPS tracking data may not. The NAB dataset incorporates a wide variety of metrics, from IT CPU and network utilization to industrial machine sensors, from web servers to social media activity. The dataset also includes data files without anomalies.

Third, we specified that the dataset must include common challenges when detecting anomalies in streaming data, including noise or establishing new patterns.

The Scoring Mechanism

At a high level, an anomaly detector is doing two things: accepting data input and flagging instances that it finds to be anomalous. Traditional scoring methods give credit when an anomaly is correctly identified but not for early detection. It’s clear that these methods don’t work well for anomaly detection in streaming data, but how should we score anomaly detections? In creating NAB, we first defined the ideal anomaly detector as one that:

1. Detects all anomalies present in the streaming data
2. Detects anomalies as early as possible (and ideally before human detection)
3. Triggers no false alarms (false positives)
4. Works in real-time (without looking ahead)
5. Is fully automated across all data sets (doesn’t require any human intervention)

NAB introduces a new scoring mechanism, one designed to reward detection algorithms that display the above characteristics. The scoring mechanism contains three key components: anomaly windows, a scoring function and application profiles.

Anomaly windows

With the goal of incorporating the value of timely detections, NAB uses anomaly windows, which are defined ranges of data points that surround each labeled anomaly instance. It uses these windows to identify and assign weights to true positives, false positives and false negatives. The benchmark gives credit to the first true positive found within the window and ignores any subsequent true positives within the window. If a detection is found outside a defined anomaly window, this is a false positive and it receives a negative score. An empty window is a false negative and receives a negative score as well. The windows are defined to be large enough to count early detections as true positives but not so large that they invite random and irrelevant detections.

Scoring Function

The scoring function is directly tied to the anomaly window. You can think of the function assigning positive or negative points to an algorithm's detections. Positive points are given for detection within a window while negative points are given for detection outside a window. The function incorporates the value of time by assigning more points to detections earlier in a window. Similarly, detections that occur slightly after the window are penalized less than detections that occur well outside of it.[2]

Application Profiles

Just as anomalies can vary across domains and applications, the value of false positives and false negatives can, too. Consider a scenario where a hospital is monitoring EKG data on a patient. A false negative (a missed anomaly) could result in catastrophic heart failure. A false positive on the other hand could simply result in sending a nurse or doctor to check on the patient. In this case, a false negative is much more costly than a false positive – potentially even fatal! Consider a different scenario where a datacenter is monitoring individual servers. In this case, false negatives are not that bad; the system is designed to be fault tolerant so the occasional failure may have little or no impact on the overall system. An abundance of false alarms, on the other hand, could cause significant disruption.

NAB accounts for these differences by including three different application profiles: standard, one that rewards few false positive detections, and one that rewards few false negative detections. The NAB code is designed such that you can easily adjust the relative weights to these preferences based on your own sensitivity.

[2] For details on the math behind the scoring function, see A. Lavin and S. Ahmad, "Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark," in *14th International Conference on Machine Learning and Applications (IEEE ICMLA'15)*, 2015. http://arxiv.org/abs/1510.03336

This combination of anomaly windows, scoring function and application profiles enables NAB to evaluate real-time performance of anomaly detectors, incorporate the value of timely detections into the standard classification metrics (true and false positives and negatives) and account for different use cases.

Community-Driven

NAB was created to be a community tool that will benefit researchers in academia and industry. As such, it is an open source code base that allows for complete visibility into the benchmark code and data files. Anyone can access the code repository to view the data, algorithms and documentation. Users can clone the repository for additional experimentation and submit changes through pull requests. NAB was built through collaboration with the community and will continue to evolve with the community as more people contribute data and anomaly detection algorithms. As we add more data files to the dataset, we plan to follow a documented versioning process. Posted scores from contributed algorithms will continue to reflect a specific NAB version number.

Results

NAB v1.0 includes results from three open source and commercially used algorithms: Numenta's HTM detector, Etsy's Skyline and Twitter's AnomalyDetection. For more information on these algorithms see the appendix.

The final NAB score sums the scores across all 58 data files, yielding a numerical result on a scale of 0 to 100 where 0 is an algorithm that makes no detections and 100 is one that is perfect, i.e. identifies all the anomalies with no false positives.

TABLE 1. NAB SCOREBOARD

Detectors	**Scores for Application Profiles**		
	Standard	***Reward low FP***	***Reward low FN***
1. Numenta HTM	64.7	56.5	69.3
2. Twitter ADVec	47.1	33.6	53.5
3. Etsy Skyline	35.7	27.1	44.5
4. Random	16.8	5.8	25.9
5. Null	0.0	0.0	0.0

Each algorithm has its own strengths and weaknesses. All algorithms' scores decreased from the Standard profile to the Reward Low FP profile, and increased for the Reward Low FN profile. A closer look at the results brings some interesting details to light.

Figure 1 shows CPU usage on a production server over time and contains two anomalies, as shown by the red anomaly windows. The first anomaly is a spike in usage, which all three algorithms detected. The second is a change in the usage pattern. HTM (represented by a diamond shape) and Skyline (the square) demonstrate the value of continuous learning by detecting this change and then quickly recognizing it as a new normal. In this case, Skyline learns the new normal quickly, HTM has a few false positives and then learns, while Twitter (the plus sign) does not identify the new pattern and continues to generate anomalies well past the window (false positives). Skyline would receive the highest score for this data file, with HTM next and Twitter last.

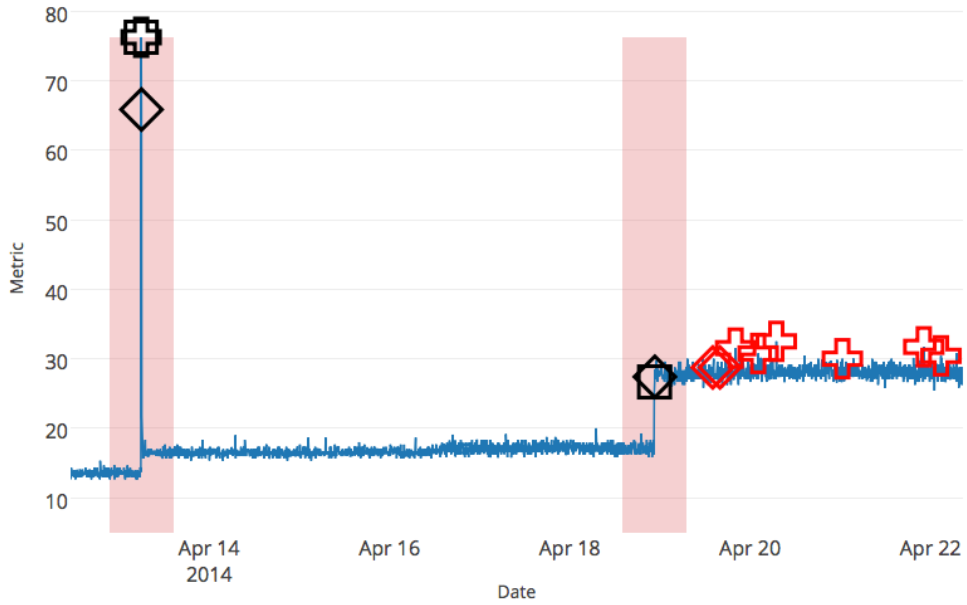

Figure 1. Detection results for anomalies on a production server based on CPU usage. A black shape indicates the first true positive detected. A red shape indicates a false positive. The red shaded regions denote the anomaly windows.

Figure 2 measures machine temperature readings. In this example, there are also two anomalies. The second anomaly represents a massive system failure, which all three algorithms detected. The first anomaly, however, is a temporal anomaly that was only detected by HTM. This anomaly is much more subtle, because the individual values in this window are all within the expected range. It's the behavior, or the temporal pattern of those values, that is anomalous. In this example, both HTM and Skyline also had a false positive.

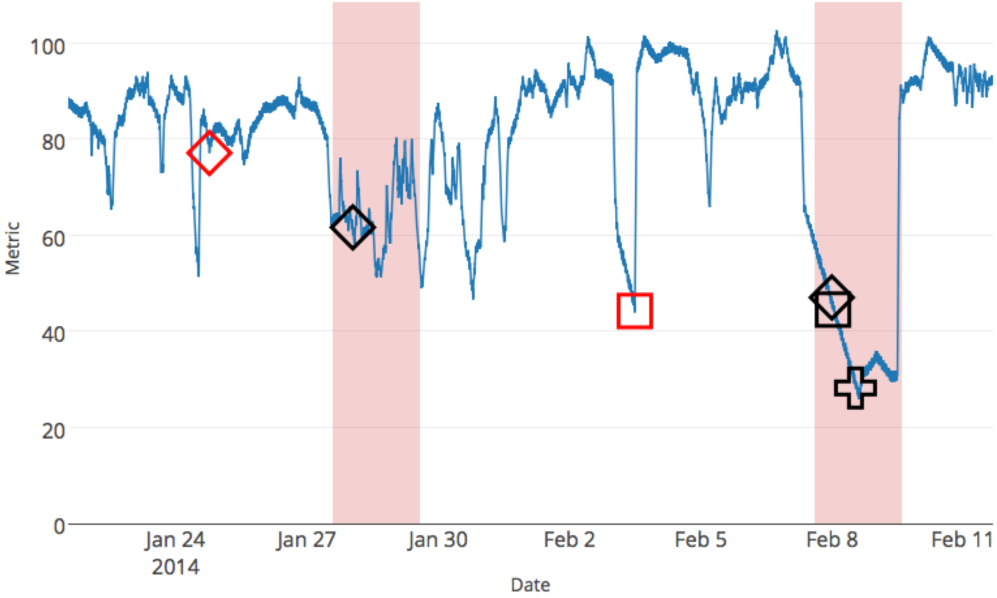

Figure 2. Machine temperature readings

In the final example, Figure 3 shows a scenario where all three algorithms detect the anomaly, but HTM detects it three hours earlier. This and the previous example illustrate how subtle, temporal changes in behavior often precede largely detectable, visible anomalies. The ability to detect these subtleties earlier than with traditional techniques opens the door to increased early warnings and prevention of unwanted outcomes, failures or catastrophes. The NAB dataset includes examples of these precursor anomalies by design, and the scoring scheme rewards algorithms for making these early detections.

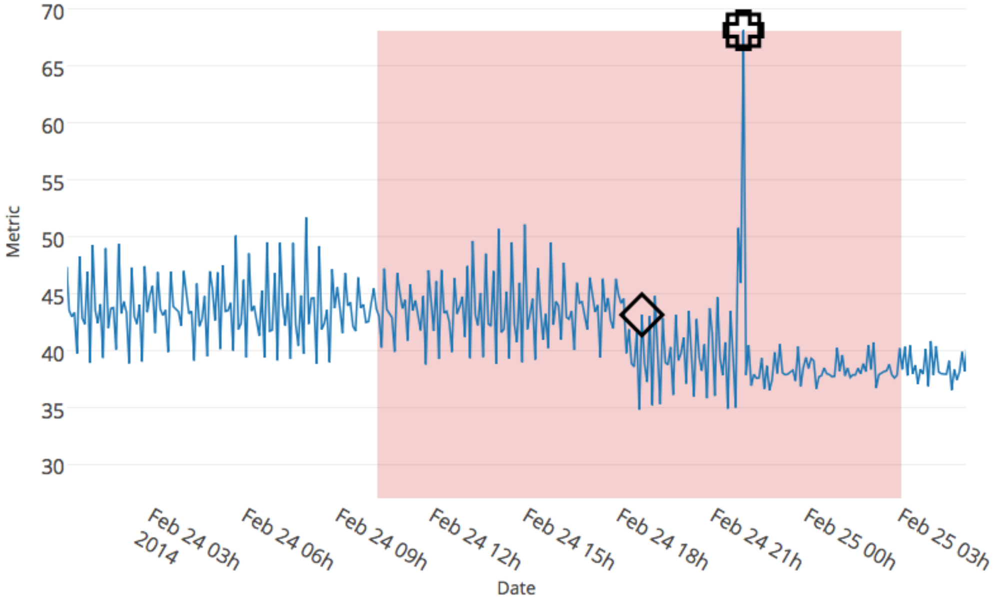

Figure 3. Early detection of machine temperature change

Conclusion

The creation of NAB marks the first temporal anomaly benchmark to provide a controlled and repeatable environment for measuring and comparing different real-time anomaly detection algorithms. NAB consists of three important and distinct features:

- Benchmark dataset – a rare and valuable collection of real-world time-series data files across multiple domains, labeled with anomalies
- Scoring function – an adaptation of traditional methods that incorporates time and rewards early detection
- Open source code – a fully open source repository containing data, algorithms and documentation

With such a benchmark in place, we can begin to understand the strengths and weaknesses of numerous anomaly detection algorithms. Over time we hope researchers will use NAB, share their results and develop new anomaly detection algorithms designed specifically for real-time streaming applications.

If you are interested in contributing data or have questions about NAB, contact nab@numenta.org.

Appendix

Algorithms[3]

Numenta HTM Detector

This algorithm is based on HTM, or Hierarchical Temporal Memory,[4] which is a model for understanding the neocortex. It uses a memory-prediction framework where in any real-time data stream, it models the sequences in the stream and makes multiple predictions for the next value. It compares each prediction to the actual value to determine an anomaly score. If a prediction is significantly different, it receives close to a 1, if it's exactly the same, it receives a zero. Continuously performing this action leads to the final detection of an anomaly.

This algorithm lends itself nicely to real-world streaming data. It can handle both predictable and highly unpredictable data, because it's making multiple predictions and continuously learning. It also doesn't require retraining or manual intervention. The code for this algorithm is available in open source at www.numenta.org.

Etsy Skyline

Skyline was developed by Etsy.com to monitor its high traffic e-commerce website. The algorithm uses a mix of popular approaches, including a set of simple detectors that measure deviation from specific variables and a voting scheme to determine the final score. Skyline is also well suited for analyzing streaming data across a wide range of applications. The code is open source and has been used in commercial settings.

Twitter AnomalyDetection

Twitter has released two versions of a real-time anomaly detection algorithm that use a combination of techniques, statistical metrics and piecewise approximation to uncover long term trends. We've scored AnomalyDetectionVec, which is intended for general usage in data without timestamps.

In addition to the three open source algorithms, NAB includes a "null" algorithm that is used as a baseline to scale all other algorithms and a "random" algorithm that provides insight into the possibility of chance high scores.

[3] For details on the algorithms evaluated, see A. Lavin and S. Ahmad, "Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark," in 14th International Conference on Machine Learning and Applications (IEEE ICMLA'15), 2015. http://arxiv.org/abs/1510.03336

[4] For more on HTM, or Hierarchical Temporal Memory, see J. Hawkins and S. Ahmad, "Why Neurons Have Thousands of Synapses, A Theory of Sequence Memory in Neocortex" arXiv:1511.00083v1 [q-bio.NC]. http://arxiv.org/abs/1511.00083v1

About Numenta

Founded in 2005, Numenta has developed a cohesive theory, core software technology, and numerous applications all based on principles of the neocortex. Laying the groundwork for the new era of machine intelligence, this technology is ideal for large-scale analysis of continuously streaming data sets and excels at modeling and predicting patterns in data. Numenta has also developed a suite of products and demonstration applications that utilize its flexible and generalizable HTM learning algorithms to provide solutions that encompass the fields of machine generated data, human behavioral modeling, geo-location processing, semantic understanding and sensory-motor control. In addition, Numenta has created NuPIC (Numenta Platform for Intelligent Computing) as an open source project. Numenta is based in Redwood City, California.