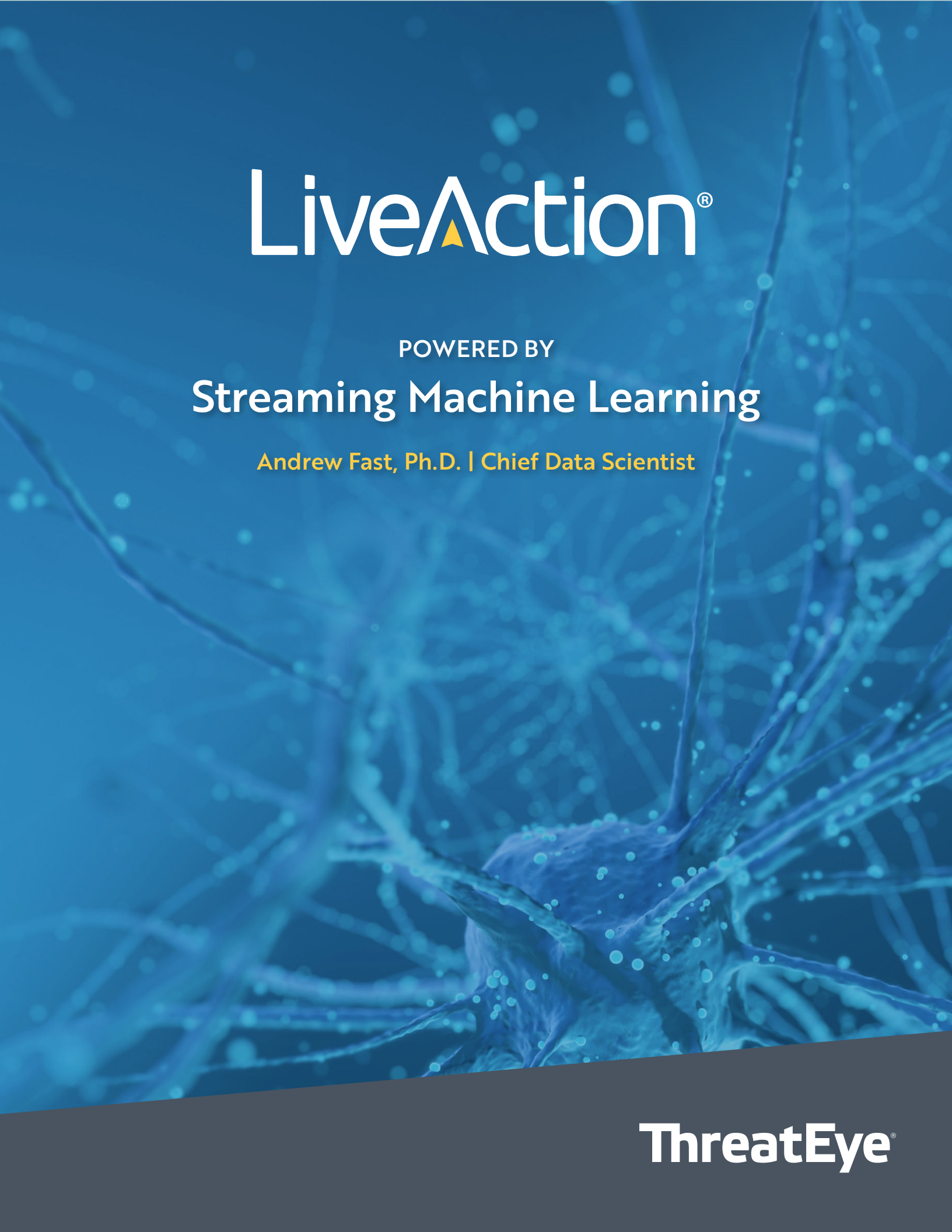

LiveAction®
POWERED BY
Streaming Machine Learning
Andrew Fast, Ph.D. | Chief Data Scientist
ThreatEye®

LiveAction®

POWERED BY

Streaming Machine Learning

Andrew Fast, Ph.D. | Chief Data Scientist

Introduction

ThreatEye employs streaming machine learning to solve the challenges facing today's network security and network operations practitioners. Two specific challenges are making this difficult task even harder. First, threat actors' tactics and techniques continuously evolve. Second, network traffic growth is showing no signs of slowing. Machine Learning (ML) would be a perfect response to changing threat environments, except that typical ML infrastructure, such as data lakes, needs to be scaled to the speed and size of network data. This level of scaling requires significant resources and is often out of budgetary reach.

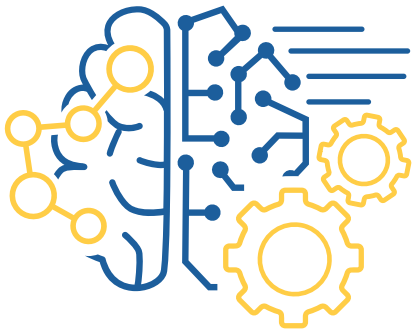
By using streaming machine learning algorithms[1], ThreatEye provides the power of machine learning at scale while maintaining reasonable resource requirements. In this paper, we unpack how streaming ML differs from other approaches. We also highlight how ThreatEye utilizes a small resource footprint while still exceeding the performance requirements of the highest bandwidth networks.

[1] Also known as incremental or online learning.

ThreatEye®

Machine Learning for Network Security

Machine learning is critical for responding to today's dynamic threat environments. Three key resources are needed for NOC and SOC groups to be successful with machine learning: Data Collection, Data Engineering, and Algorithmic Computation (for models)



1 Data Collection

involves extracting metadata directly from the network packet stream. Legacy tools have historically leveraged Deep Packet Inspection (DPI) to inspect packet payloads for useful data. Now that the bulk of network traffic is encrypted, DPI yields less and less useful data. Unlike DPI-based solutions, ThreatEye's probe software extracts Deep Packet Dynamics (DPD), a rich metadata feature set without payload inspection.

2 Data Engineering

is the process of moving the raw data to the right place and transforming it for algorithmic computation. Sometimes referred to as DataOps, data engineering includes tasks such as data standardization and feature creation. Feature creation varies widely from model to model, but often includes steps such as computing historical summaries of past data. Data engineering is critically important to ensure that the same data and features are available in production that are available during training of the model.

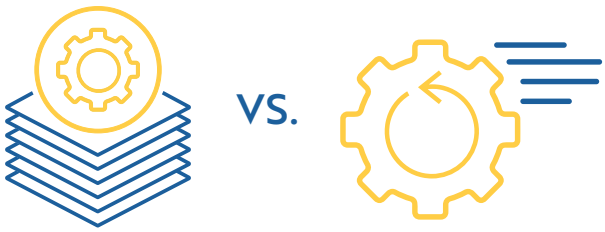
3 Algorithmic Computation

is the final stage where machine learning algorithms are applied to the data. This includes the necessary steps of training and testing models. In this phase ThreatEye employs both supervised and unsupervised ML techniques in its analysis. For network data which is constantly changing, models must be regularly updated to account for that change. This need is known as concept drift. Accounting for concept drift usually requires significant computational infrastructure to support the feature creation and model re-training over large amounts of data. Once the model is trained, it is tested for accuracy before the model is moved into production.

> **"Concept drift in machine learning and data mining refers to the change in the relationships between input and output data in the underlying problem over time."**

TWO TYPES OF MACHINE LEARNING:

Batch and Streaming



Network practitioners have at their disposal two approaches to Machine Learning:

Batch and **Streaming**. At a very high level, these two techniques differ in meaningful ways. With Batch Machine Learning, models are trained off-line using historical, retrospective data, and then are later deployed retrospectively on data that has been saved for analysis. With Streaming ML, however, models are trained incrementally as data arrives with analysis occurring in real-time. As we compare and contrast Batch vs. Streaming, it is important to note that each approach can be more or less useful addressing different use cases. Instances requiring quicker access to actionable insights may benefit more from Streaming. Alternatively, uses cases benefiting from more than one look at the data may be better served by Batch processing. Also worth noting is that traditional batch ML models can often be deployed—but not trained—within a streaming engine. For CounterFlow, specifically, ThreatEye offers real-time, encrypted traffic analysis. For this use case, a Streaming approach fits perfectly. The good news is that solving the visibility problems around encryption supports a Batch approach as well. CounterFlow's ThreatEye Recall, for example, provides line-rate full packet capture for retrospective study. As discussed, Batch processing is well suited for this type of analysis.

THE TYPICAL APPROACH TO ML:

Drowning in the Data Lake

A traditional batch approach to machine learning works something like this: first, create a highly engineered data pipeline to port all data back into a massive data lake. Next, historical features are created by running queries and pre-processing scripts. Finally, models are trained on this large collection of data.

Moving a trained model to production requires translating every data processing step that was performed in the database into an outward facing application. For batch machine learning, considerable effort is required to keep these training and production pipelines synchronized. For example, data queries to create historical features need to be re-run for each model window. Particularly for large data lakes, feature creation and model training alone require significant computing resources.

When performed during the day, these processes run the risk of depriving an organization of access to the data lake for daily business activities. In these cases, model training can be pushed to overnight. Unfortunately this shift slows the feedback loop and reduces the response time to new network behavior. In order to keep up with high-speed data, the batch approach to machine learning requires significant person hours, many machine resources and large numbers of resource-intensive model retrains.

STREAMING MACHINE LEARNING:

A New Alternative

Streaming ML is a different approach for applying machine learning on high-speed data, one with realistic person hours and minimal budgetary resources. Instead of making multiple, costly passes through the data to compute features, streaming machine learning algorithms are designed to work with only a single look at each data point. Using this approach, streaming algorithms can be applied to possibly infinite datasets without requiring large amounts of data storage. Furthermore, streaming models can respond to concept drift organically without intensive retraining. Though the overall approach is different, many of the core concepts of streaming models carry over from the batch setting. These are summarized in the table below:

CONCEPT	BATCH	STREAMING
Data Handling	Multiple passes over the complete data	One-Shot – a single look at each data point
Training Strategy	Hold-Out/Cross Validation	Progressive Validation
Concept Drift	Must retrain static model	Automatic
Data Storage	Large data storage	Small Feature Cache
Data Size	Finite	Infinite
Training Data	Training Set	Burn-In/Convergence Window
Model Pipelines	Two: Training and Production	One: can prototype and deploy using same infrastructure

Many common machine learning algorithms such as logistic regression, linear regression, decision trees and random forest have been converted to streaming algorithms by using a different optimization function. For example, many models—including logistic regression—are trained using stochastic gradient descent. These models can be transformed to streaming by using one-shot gradient descent instead. One notable exception is deep learning models. They typically require many passes over the data and have not yet been translated to the online, streaming paradigm outside of academia.

Summary

While many of the machine learning concepts and algorithms carry over from batch to streaming, the sizeable infrastructure and repeated training necessary for batch processing is no longer required. The remarkable thing is that, despite this reduction in resources, streaming algorithms like those used within ThreatEye achieve similar performance to batch algorithms with the additional benefit of responding to constantly evolving data.

LiveAction®

960 San Antonio Rd, Suite 200, Palo Alto, CA 94303
+1 (888) 881-1116

About LiveAction

LiveAction provides end-to-end visibility of network and application performance from a single pane of glass. This gives enterprises confidence that the network is meeting business objectives offers IT administrators full visibility for better decision making and reduces the overall cost of operations. By unifying and simplifying the collection, correlation and presentation of application and network data, LiveAction empowers network professionals to proactively and quickly identify, troubleshoot and resolve issues across increasingly large and complex networks. To learn more and see how LiveAction delivers unmatched network visibility, **visit www.liveaction.com.**