Lynx

AI Against Fraud & Financial Crime

Daily Adaptive Model

White Paper

lynxtech.com

Carlos Santa Cruz

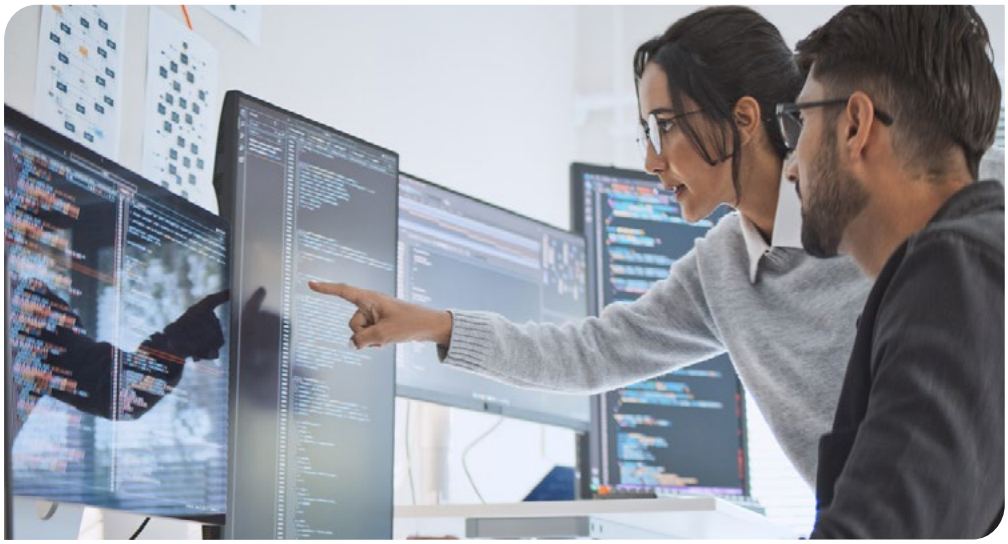
CTO
Chief Technology Officer

Carlos Santa Cruz is the **CTO of Lynx** and a **Professor of Computer Science and Artificial Intelligence at the Universidad Autónoma de Madrid (UAM).**

He has been involved in the development and design of fraud detection and anti-money laundering applications from the outset. At Lynx, he leads the creation of advanced tools that use AI to help customers identify and prevent fraudulent transactions. With a PhD in solidstate physics, Carlos has a strong foundation in mathematics, computer science, and statistics. His passion for AI stems from his belief in its potential to make a meaningful impact on the world.

Lynx

Next-Generation Fraud Detection with Lynx's Daily Adaptive Models



Executive Summary

The digital transaction landscape is dynamic, with ever-evolving payment technologies, shifting customer behaviors, and increasingly sophisticated fraud attacks. **Machine Learning (ML) models used for fraud prevention must frequently be updated to maintain effective detection.**

The traditional static ML model construction process involves several key steps:

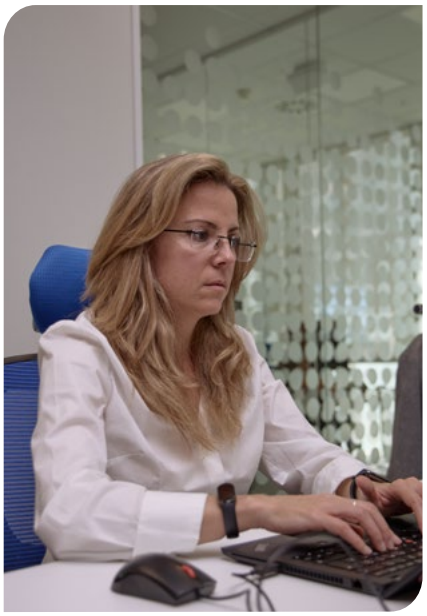
- **ML models** are trained on **historical transaction payloads** from a financial institution's payment systems.
- **Creating Training and Test Data Sets:** Historical transactions are divided into a training set, which is used to build the model, and a test set, which is used to measure its performance.
- **Information Preprocessing:** Transaction data is prepared and cleaned for model ingestion.
- **Feature Extraction:** Transactions are transformed into tens of thousands of meaningful features (characteristics) relevant to fraud.

- **Feature Selection:** The features most relevant to fraud are selected to cut computational costs and accelerate model training. This can also introduce unwanted bias.
- **Model Building with the Training Set:** The ML model ingests the selected features, building logic that determines how to best detect fraud.
- **Model Validation with the Test Set:** The ML model's performance is validated using the test data set to determine how well the model detects fraud. Model construction is repeated until the best-performing model is created and deployed in production.

Static ML models only retrain after several months, leading to performance degradation (model drift) over time due to constant changes in the transaction and fraud environment, which leads to increased fraud losses and higher false positive rates.

Lynx's Daily Adaptive Models (DAMs) update daily to incorporate the latest payments, user behaviors, and fraudster attack patterns, maintaining model performance. DAMs also improve upon traditional ML model building in two critical ways:

- **Feature selection is performed via Lynx's proprietary training algorithm to avoid unwanted bias,** unlike most model training approaches, which perform feature selection before model training.
- **Lynx enriches transactions with feeder data from customer onboarding and applications, device intelligence, behavioral biometrics, and more.** This adds valuable transaction context for more nuanced detection and a complete picture of each customer's financial behaviors.



Most ML models are limited to analyzing transactions that match the original transaction payload type on which they were trained. They are unable to adapt to new types of data and payment channels. **This is an issue in the dynamic payment system environment, where financial institutions build products, acquire technologies, and change customer portfolios.**

Executive Summary

Lynx Flex enables dynamic payloads for multi-channel coverage and data extensibility; as new channels and data types are added, new features are generated, and DAMs are retrained to enable optimal fraud detection. Lynx Flex propagates the new features throughout Lynx's workflows, dashboards, and reports, giving FIs greater visibility over their fraud and payment environment.

DAMs and Lynx Flex work together to give FIs superior fraud detection:

- Higher detection accuracy and less drift
- Flexibility and extensibility at scale
- More savings with fewer false positives
- Lower operational costs and reduced alert fatigue
- Improved customer experience and lower turnover
- Improved brand image.

Lynx's customers stop more fraud and can save millions each year with DAMs, with an average 80% Value Detection Rate (VDR) at 3-5 false positives per 10,000 transactions.

Website: lynxtech.com
Email: info@lynxtech.com

Introduction

Digital transactions constantly change as companies develop new products and services, payment methods evolve, and users change their behaviors. At the same time, new fraud patterns emerge as criminals adapt their attacks to bypass fraud prevention efforts.

Machine Learning (ML) models used in fraud prevention must be updated frequently to capture these dynamic factors and catch the most fraud possible. They also must be able to ingest new data types to account for emerging products, technologies, and payment types relevant to fraud. Models that don't keep up with the shifting transaction and fraud environment drift and degrade over time.

Lynx's Daily Adaptive Models (DAMs) combine daily retraining and data extensibility to drive increased fraud savings and reduced friction for financial institutions. This white paper describes how Lynx updates its ML models daily to prevent more fraud than static ML models commonly used in fraud prevention. The benefits of incorporating dynamic payloads with Lynx Flex, which introduces the ability to configure API and intelligence feeds in a user interface to provide data extensibility that propagates to models, rules, and reports, are also discussed. First, however, essential concepts in machine learning and static models will be explored.

A Static ML Model Training Procedure

Many financial institutions and fraud prevention solutions rely upon static ML models to detect fraud. Static models are trained once and then used for an extended period of time in production. This legacy approach is limited, given the fast-paced changes in transaction fraud. To understand why, it is important to first review how static models are built, trained, and deployed.

Sample Transaction Payload

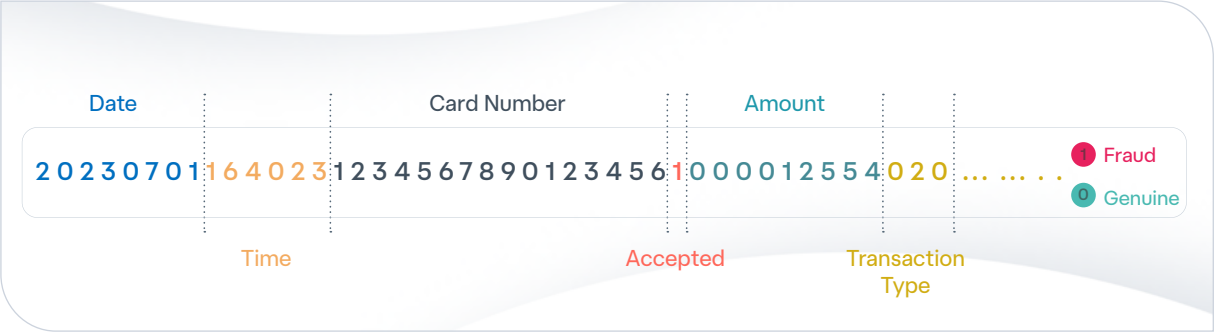


Figure 1

Transaction Payloads

ML models are built (trained and tested) using historical transaction payloads from a financial institution's payment systems. ***Figure 1*** is a simplified example of a card transaction payload and shows the fields involved in a card operation, such as transaction type, date, time, card number, amount, and whether the transaction has been accepted. There is also a label* indicating if the transaction was fraudulent or genuine (red or green, respectively). Fraud and Genuine labels are derived from financial institutions' analysts who determined whether the transactions were fraudulent or genuine.

*Note: The use of labels in ML model training is referred to as Supervised learning while training without labels is referred to as Unsupervised learning. For a deeper discussion of this distinction and the benefits of Supervised Learning in fraud detection, see the section titled 'Supervised vs. Unsupervised Learning' in Lynx's white paper ***Unlock Real-Time Fraud Detection with Supervised Machine Learning.***

Training Sets and Test Sets

The available historical transactions are divided into sets of data for model training and testing. The transactions within this time period are divided into two sets of data: a training set, which will be used to build the model, and a test set, which will be used to measure its performance.

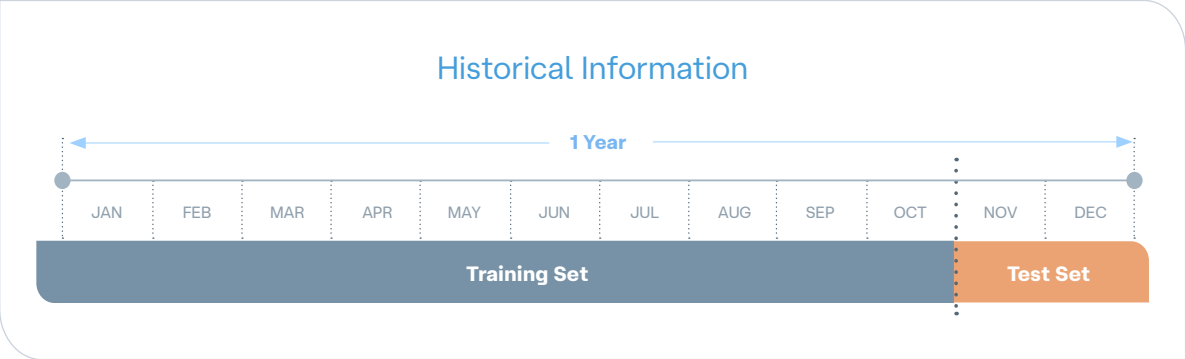


Figure 2

Figure 2 shows a financial institution's historical transactions over a one-year period. In this case, the training set consists of all transactions from January through October, while the test set consists of all transactions from November through December. The specific time periods and data selected for each set can vary, but model building always uses both a training set and a test set, and the training set will always contain older data than the test set. The training and test sets are deliberately separated to maintain objectivity; once built, the model's performance will be measured against data it has not ingested during training.

Now that transaction payloads and the development of training and test data sets have been described, the model-building process can be explored. ***Figure 3*** provides a high-level overview of the typical process for building static ML models.*

*Note: for an in-depth discussion of ML model construction and validation, refer to Lynx's white paper ***Unlock Real-Time Fraud Detection with Supervised Machine Learning.***

Information Preprocessing

The preprocessing phase involves cleaning transaction data and parsing transactions into fields to prepare the data for ML model ingestion during training. This is an important opportunity to identify issues with the available data, as low-quality or unreliable data (such as data fields that are incomplete or in the wrong format) can negatively affect model accuracy if used in training. Both the training and test data sets are preprocessed.

As previously shown, ***Figure 1*** displays a segment of a preprocessed transaction that has been transformed into numbers and separated into fields (date, time, card number, etc.).

Feature Extraction

Once transaction information has been preprocessed, feature extraction is performed to transform the original transactions into a set of features that characterize the transactions. Feature extraction can be likened to a game of "Guess Who" where a feature extraction algorithm asks many questions about the data to create new features.

For example, has the customer ever made a transfer to this beneficiary? What is the typical transaction value? How many transfers do they make per day, week, or month? How many devices do they use? Has the device changed? Thousands of these kinds of questions are asked to extract thousands or tens of thousands of new features from the transaction data. An example of an extracted feature might be "number of transactions in the last 2 hours."

Figure 3

After feature extraction, each transaction is represented by a new set of features (called a feature vector) along with its Fraud or Genuine label. ***Figure 4*** displays an example of a simplified feature vector: amount and number of transactions in the last 2 hours. Since the feature vectors describe the transactions and the labels identify whether they were fraudulent, they can now be used to train the ML model to effectively detect fraud.

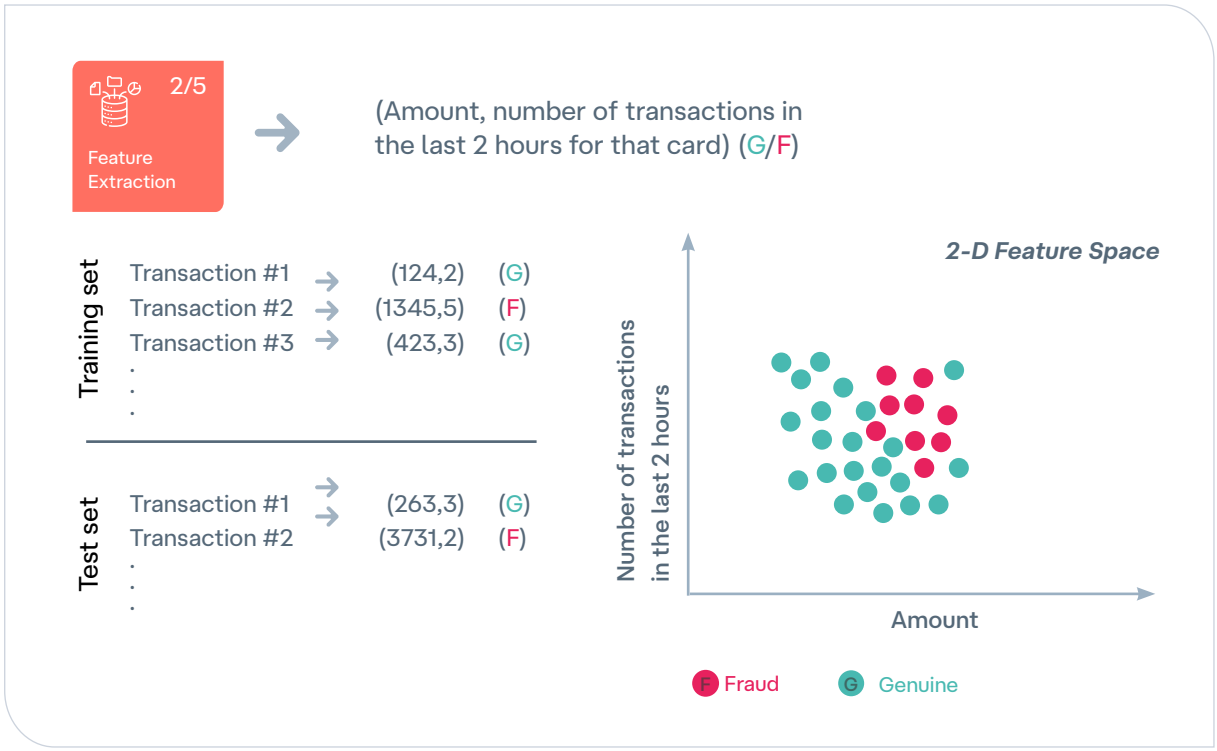


Figure 4

Feature Selection

In general, hundreds, thousands, or tens of thousands of features may be generated during feature extraction. A subsequent feature selection phase is typically performed with feature selection algorithms that choose features that are most relevant for identifying fraud and eliminate those that are redundant or irrelevant.

Feature selection processes, although costly up front, can significantly reduce dimensionality, accelerating subsequent model training and reducing computational costs. However, these processes may also introduce unwanted bias into the model, as selecting and eliminating features before training influences which features the model trains on and which it doesn't.

An alternative approach is to allow the model's training algorithm itself to ingest all extracted and transformed features and select those that are relevant to the problem. Lynx has developed a fast and efficient training algorithm that chooses the most important features to minimize dimensionality, reduce training costs, and significantly reduce bias. This improves model performance and allows the model to adapt to changing fraud patterns. This is explained further below in the section titled "Daily Training."

Model Building

Once features have been extracted and selected, they are ingested by the model during training. The ML model essentially builds logic around the extracted and selected features to solve the problem at hand- detecting fraud.

The model's training algorithm is applied to the training data set (composed of the extracted and selected features) to calculate the model parameters (internal model variables learned from the training data that allow the model to make predictions). Various types of training algorithms are available, ranging from artificial neural networks to ensembles of decision trees.

Model Validation

Once the model has been trained, its performance is evaluated against the test data set not used during model training. This is a crucial step that ensures that the model can accurately identify fraud in new data it hasn't seen before.

The model's performance is measured using two key metrics: the Value Detection Rate (VDR) and the Transaction False Positive Rate (TFPR). The VDR indicates the proportion of actual fraud detected (accuracy), which directly correlates with potential cost savings. TFPR indicates the rate of false positives- genuine transactions incorrectly identified as fraudulent.

Figure 5

The model's performance is subsequently visualized in a performance graph with a Receiver Operating Characteristic (ROC) curve, depicted in ***Figure 5.*** The better a model performs—indicating a higher VDR and a lower TFPR—the more the ROC curve approaches the graph's upper left corner.

To determine if the model's performance is the best possible outcome, the cycle of feature extraction, feature selection, model training, and model validation is then repeated many times with different combinations of features and training algorithms. Finally, the best-performing model is deployed in the live production environment, which scores incoming transactions for fraud over the next several months (typically 6 months).

Retraining

After deployment, the static model-building process begins again using data from new transactions that have occurred since the model was last trained. ***Figure 6*** shows a typical timeline for the static machine learning model retraining process.

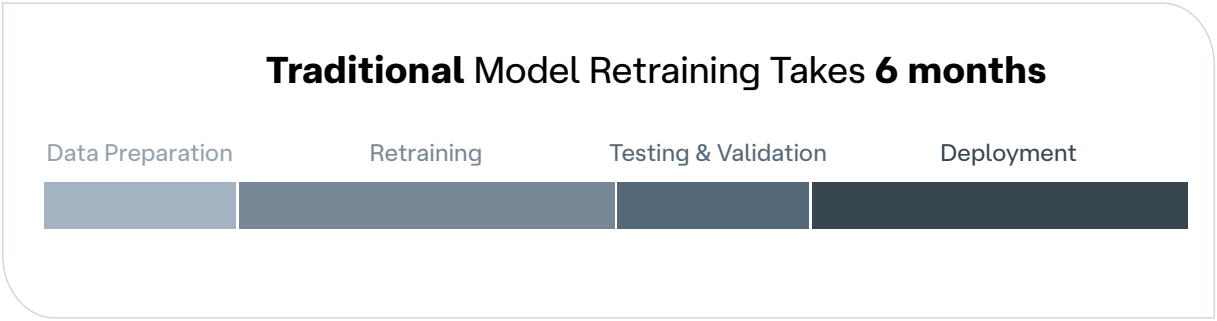


Figure 6

Limitations of Static Training in a Dynamic Fraud and Transaction Environment

Humans and technologies are fundamentally dynamic, and customer transaction behaviors, technologies, and fraudster attack patterns constantly change. For the purposes of fraud detection, it is unreasonable to expect that transaction data from a specific time period will continue to be representative of all past and future transactions.

However, this is precisely what static ML models do. As these models remain in production unchanged for months at a time, the transaction data they evaluate becomes more and more differentiated from the training data they were built around. Inevitably, the models drift and deteriorate as they await retraining, resulting in:

- Unidentified new and emergent fraud cases, which lead to losses that compound as criminals double down and exploit model weaknesses
- Lower detection accuracy, allowing more successful fraud and creating more false positives (genuine transactions scored as fraudulent)
- Increased operational costs to manage fraud impacts and false positive alerts
- Diminished user experience as false positives incorrectly block genuine activities
- Increased customer turnover as customers become frustrated with a poor user experience
- Brand and reputational damage due to more cases of fraud and more false positives

A Daily ML Model Training Procedure

Lynx has developed daily adaptive models that retrain every day on new data obtained from transactions and reported fraud.

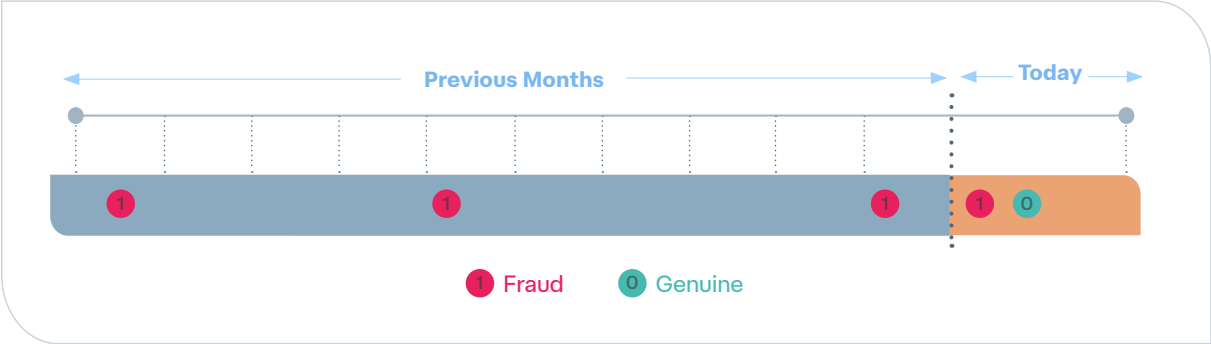


Figure 7

Figure 7 illustrates how Lynx's DAMs receive transactions and reported fraud cases. Note that the new fraud labels may correspond to transactions made today (such as the red fraud case on the far right of ***Figure 7***) but may also correspond to transactions made days, weeks, or months ago. Additionally, Lynx adds intelligence through feeder data (customer onboarding, application, and identity verification data such as the customer's salary and travel frequency) to enrich transaction payload data, creating additional features and a more complete picture of each customer's financial behavior around current and prior transactions.

Figure 8 shows the retraining procedure Lynx's DAMs follow each day.

After preprocessing, features are extracted from the new transaction and fraud data (feature extraction). Next, Lynx retrains the models using these new features. As discussed previously, Lynx's training algorithm selects features that are relevant for identifying fraud - contrasting with the legacy approach that uses a feature selection algorithm prior to training and adds undesired bias. Critically, this minimizes dimensionality, reduces training costs, and significantly reduces bias while improving model performance over time and ensuring model adaptability as fraud patterns evolve. Lynx also utilizes dropout, a regularization procedure that increases model robustness and generalization to new unseen data.

Since the training algorithm itself selects features that are based on new transaction data, greater importance may be assigned to features that were previously discarded if they are now more relevant to cases of fraud. The model then adjusts its parameters accordingly. This approach ensures the model selects features that are representative of both historical and recent transaction activities and captures all current and previously reported fraud. Lynx's daily retraining creates a virtuous cycle to maintain model performance using both new and old features and fraud labels.

As a final step, the updated model's fraud risk scores are assessed against the distribution of scores produced by the previous day´s model. This ensures that performance has not decreased and verifies that the model has been appropriately updated to combat drift, bias, improper score distribution, and other issues arising from new transaction and fraud data. The retrained model is then deployed to production. The entire retraining process takes just hours.

The benefits of daily training compared to static training include:

- Higher model detection accuracy and less drift, preventing more fraud and generating fewer false positives
- Lower operational costs to manage fraud impacts, incident recovery, and false positive alerts
- Reduced alert fatigue, which promotes greater talent retention, lower turnover, and lower recruitment and training costs
- Improved user experience due to fewer genuine transactions being blocked
- Lower customer turnover thanks to an improved user experience
- Improved brand image and reputation due to better fraud prevention with less unnecessary friction

Figure 8

Daily Training with a Dynamic Payload

Most static and daily ML models are limited to analyzing transactions that match the original transaction payload type on which they were trained. These models cannot adapt to new types of data and payment channels (and therefore generate associated features), which may be relevant to preventing fraud. This is an issue in the dynamic payment system environment, where new data fields and payment channels often emerge as technologies and products evolve.

Lynx solves these issues with Lynx Flex, which enables DAMs to train with dynamic payloads.

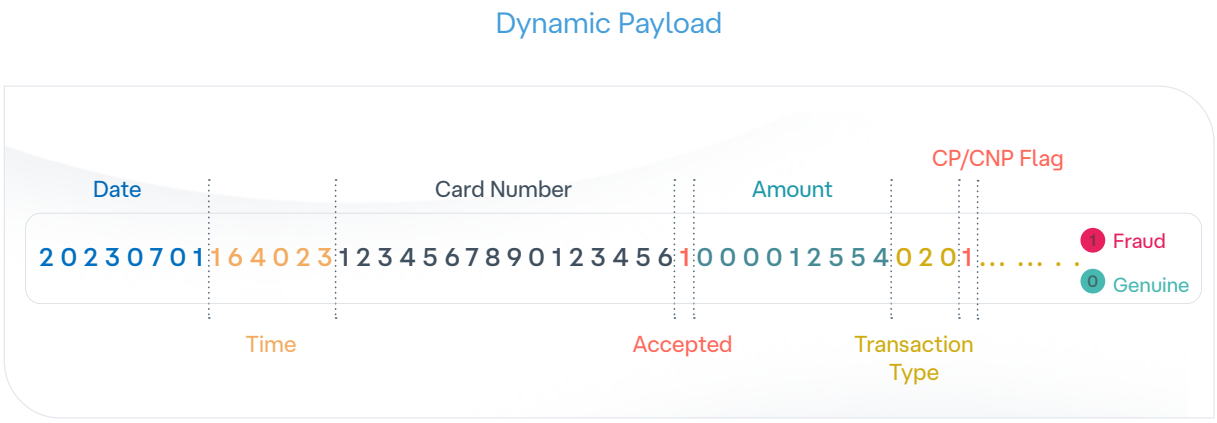


Figure 9

Figure 9 shows a transaction with a new field: Card Present/Card Not Present (CP/CNP) Flag. In this example, assume that the CP/CNP Flag field has become available because the bank providing the transaction data recently switched to a new banking application, enabling more fields to be provided.

Lynx Flex allows the new data field (CP/CNP Flag) to be incorporated throughout the entire model construction and testing procedure, including the preprocessed transaction payload, the extracted features, model training, model validation, and retraining. Lynx Flex also propagates new relevant features throughout Lynx Fraud Prevention's workflows, dashboards, and reports so financial institutions gain greater visibility and are better equipped to prevent fraud.

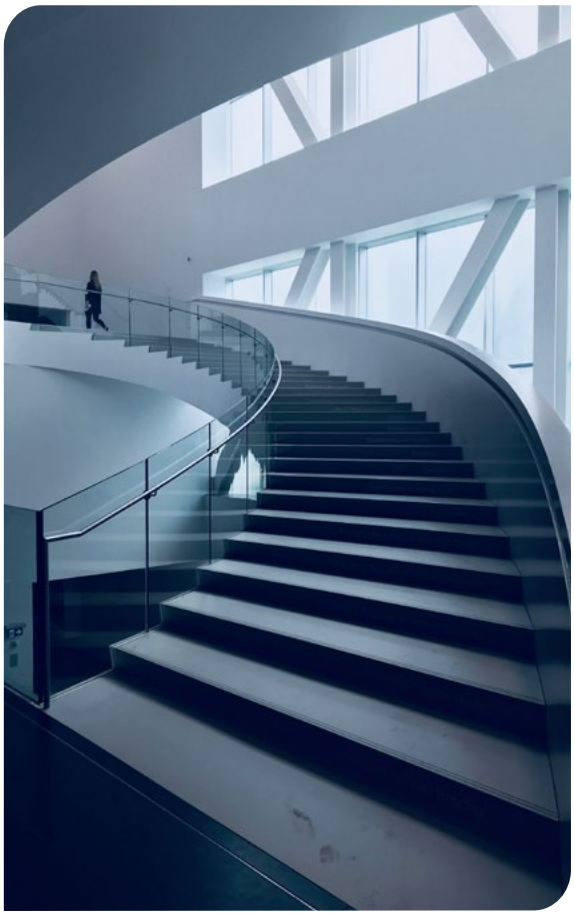
This extensibility ensures Lynx's DAMs learn and improve from any new data field or type. As new data is added, new features are generated, and the DAMs are retrained to enable optimal fraud detection. This significantly improves performance and prevents more fraud, reduces false positives, and enables more genuine and frictionless customer journeys compared to models relying on non-dynamic payloads.

Lynx's Daily Adaptive Models Outperform Static Models

Lynx's Daily Adaptive Models are a pillarstone for any financial institution. Unlike static and inflexible ML models, which drift, deteriorate, and can't learn from new payment methods or data fields, DAMs meet the changing transaction and fraud environment to remain relevant each day.

The proof lies in the outcomes Lynx has achieved for its customers, which include:

- The ability to identify and thwart new fraud attacks
- Automatic adaptations to significant data drift, such as during the COVID-19 pandemic
- An average 80% Value Detection Rate (VDR) at 3-5 false positives per 10,000 transactions
- Fraud savings of up to $500 million per year for Tier 1 banks
- $1.6 billion in gross fraud savings globally on a rolling 12-month basis (across all customers)



Key Takeaways

Static Models Underperform in a Dynamic Transaction and Fraud Landscape

- The digital transaction landscape is dynamic. ML models used in fraud prevention need to update frequently.
- Static models only update every few months and degrade over time, leading to increased fraud losses and false positives

Lynx's Daily Adaptive Models Update Daily

- Lynx's Daily Adaptive Models (DAMs) retrain daily, incorporating the latest transaction and fraud patterns for sustained high performance.
- DAMs improve upon traditional ML model building by:
  - Performing feature selection with Lynx's proprietary training algorithm to avoid unwanted bias.
  - Enriching transactions with feeder data for behavioral context and more nuanced detection.

Lynx Flex Unlocks Multichannel Coverage

- Lynx Flex enables dynamic payloads, allowing FIs to adapt to new data types and payment channels for comprehensive fraud coverage.
- New features propagate throughout Lynx's workflows, dashboards, and reports, giving FIs greater visibility over their fraud and payment environment.

DAMs and Flex Equal Superior Fraud Detection

Lynx's customers achieve significant fraud detection savings with DAMs and Flex, with an average 80% Value Detection Rate (VDR) at just 3-5 false positives per 10,000 transactions.

Discover how

Lynx's DAMs create value for organizations and their customers.
Contact us Today:
info@lynxtech.com

Lynx

About Lynx

Lynx utilizes advanced AI for fraud prevention, honed over 25 years. Originating from the Autonomous University of Madrid data science program, Lynx is trusted by leading financial institutions globally to significantly reduce fraudrelated losses. Processing over 66 billion transactions annually, Lynx's AI-driven approach illuminates real-time risks and empowers organizations to focus on crucial tasks.

Contact Us

Get in touch with us:

Website: lynxtech.com

Email: info@lynxtech.com

Recognitions

Chartis Financial Crime and Compliance50 2024

Chartis RiskTech Quadrant® Category Leader Enterprise Fraud Solutions, 2024

Chartis RiskTech Quadrant® Category Leader Payment Fraud Solutions, 2024

Chartis RiskTech Quadrant® Best of Breed Name and Transaction Screening Solution, 2024

Recognised as a Representative Vendor in the 2024 Gartner® Market Guide for Fraud Detection in Banking Payments.

datos INSIGHTS

CBINSIGHTS Fintech 100 2024

* Gartner, Market Guide for Fraud Detection in Banking Payments, 11 December 2024. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved. Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.