

第 14 章 CHAPTER 14 設計異常偵測 Designing for Anomaly Detection

在本章中，我們將解決 AI 可以偵測到的各種類型的異常，並提供針對這些使用案例設計 UI 的建議。掌握本章將使使用者體驗設計師和產品經理能夠與資料科學和工程同事進行高品質、詳細的對話，並討論介面設計的重要考慮因素。本章也探討微調系統的 UI 最佳實務，以最佳化可用性並避免誤報和誤報。

In this chapter, we tackle various types of anomalies that AI can detect and provide recommendations for designing the UI for those use cases. Mastering this chapter will enable UX designers and product managers to have high-quality, detailed conversations with their data science and engineering colleagues and discuss important considerations of interface design. This chapter also explores UI best practices for fine-tuning your system to optimize usability and avoid false positives and false negatives.

為什麼檢測異常很重要？

Why Is Detecting Anomalies Important?

異常偵測適用於各種使用案例：

Anomaly detection is useful for a wide range of use cases:

Identification of Critical Production Issues: A sudden and significant drop in signal strength at a specific telecommunications tower likely points to a critical issue, such as equipment failure. Anomaly detection helps engineers catch the problem early and fix it before it impacts the quality of service.

Quality Control and Assurance: In manufacturing (particularly in Six Sigma shops), an anomaly in a gadget's measurement on a production line may signal a quality control issue. Identifying this anomaly helps the manufacturer identify the source of the problem, improve the manufacturing process, and ensure the production of high-quality gadgets.

Security and Fraud Detection: In the financial industry, the sudden use of a credit card for multiple high-value transactions in different countries within a short time frame can indicate fraudulent activity. Early detection of such anomalies alerts the bank to block the card to decrease the impact of the liability.

and stop the hack.

Early Warning System: Anomalies play a crucial role in predictive maintenance for industrial machinery. For example, an unusual increase in vibration or temperature readings from a specific component of an industrial machine may indicate impending failure. Detecting anomalies early allows maintenance teams to schedule timely repairs or replacements, preventing unexpected downtime.

Improving Decision-Making: In an e-commerce platform, a sudden surge in website traffic beyond normal patterns during a specific time period (e.g., due to a marketing campaign) can be considered a “ happy ” anomaly. Successfully recognizing this anomaly allows the marketing team to adjust strategies in real time to capitalize on the increased interest and potentially boost sales.

Compliance and Regulation: The pharmaceutical industry ’ s manufacturing process is often particularly rigorous to help avoid product contamination. Detecting anomalies in the process (such as longer chemical reaction times) is crucial to complying with regulations. Identifying and addressing these anomalies ensures the company meets quality and safety regulations and avoids expensive fines and lawsuits.

雖然人工智慧可以檢測許多不同類型的異常，但本書將重點放在各行業廣泛認可的四種主要類型。設計 UI 以偵測這四種類型的異常的考量在複雜的使用案例中廣泛適用且具有指導意義。

While AI can detect many different types of anomalies, this book will focus on the four main types widely recognized in various industries. The considerations of designing the UI to detect these four types of anomalies are widely applicable and instructive in complex use cases.

四種主要異常類型

Four Main Anomaly Types

雖然對所有各種異常類型的完整分析遠遠超出了本書的範圍，但我們可以從安德魯·馬奎爾（ Andrew Maguire ）的異常分類（ 1 ）中汲取靈感，以檢查四種主要類型的異常以及每種類型的介面設計注意事項（見圖 14.1 ）。

While the complete analysis of all of the various anomaly types is well beyond the scope of this book, we can draw inspiration from Andrew Maguire ’ s exceptionally useful and wise anomaly classification (1) to examine four main types of anomalies and the interface design considerations for

each (see Figure 14.1).

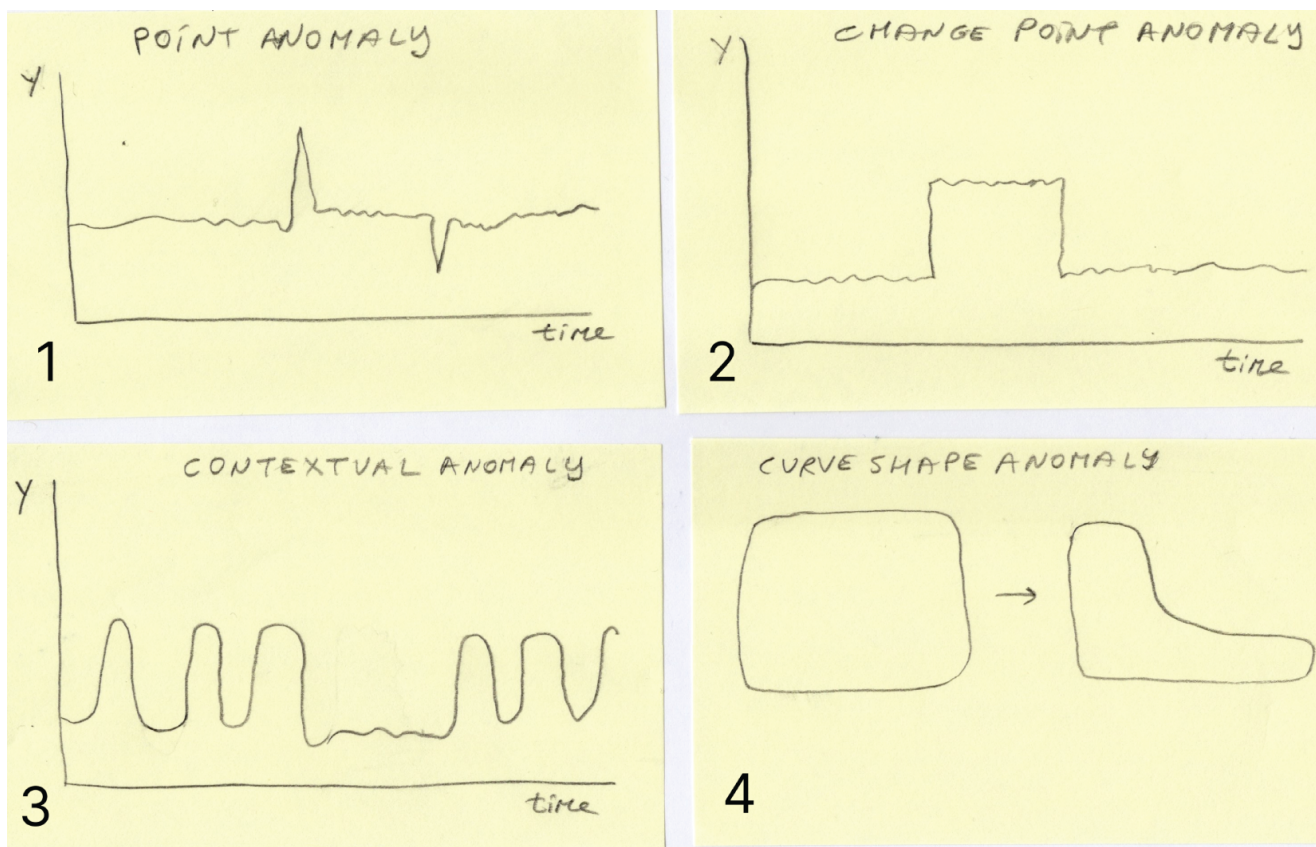


圖14.1 受Andrew Maguire啟發，AI檢測到的四種常見異常

Figure 14.1 Four types of common anomalies detected by AI, inspired by Andrew Maguire

點異常

Point Anomaly

每當值短暫「尖峰」並超過預先決定的靜態或動態閾值時，就會發生點異常。一個典型的例子可能是電腦的CPU忙碌百分比指標，該指標會因為惡意進程使用過多的處理容量而激增。

Point anomalies occur whenever the value briefly “spikes” and exceeds a predetermined static or dynamic threshold. A classic example might be a computer’s CPU Busy Percent metric, which spikes because a rogue process is using too much processing capacity.

點異常通常由在異常首次發生的周圍繪製的紅色圓圈表示。表示異常數據點讀數的紅點也是典型的（見圖 14.2）。

Point anomalies are usually indicated by a red circle drawn around the first incidence of the anomaly. Red dots indicating anomalous data point readings are also typical (see Figure 14.2).

我們如何確定什麼構成了尖峰？從廣義上講，點異常偵測分為兩類：靜態閾值和動態閾值。

How do we determine what constitutes a spike? Broadly speaking, point anomaly detection falls into two categories: static thresholds and dynamic thresholds.

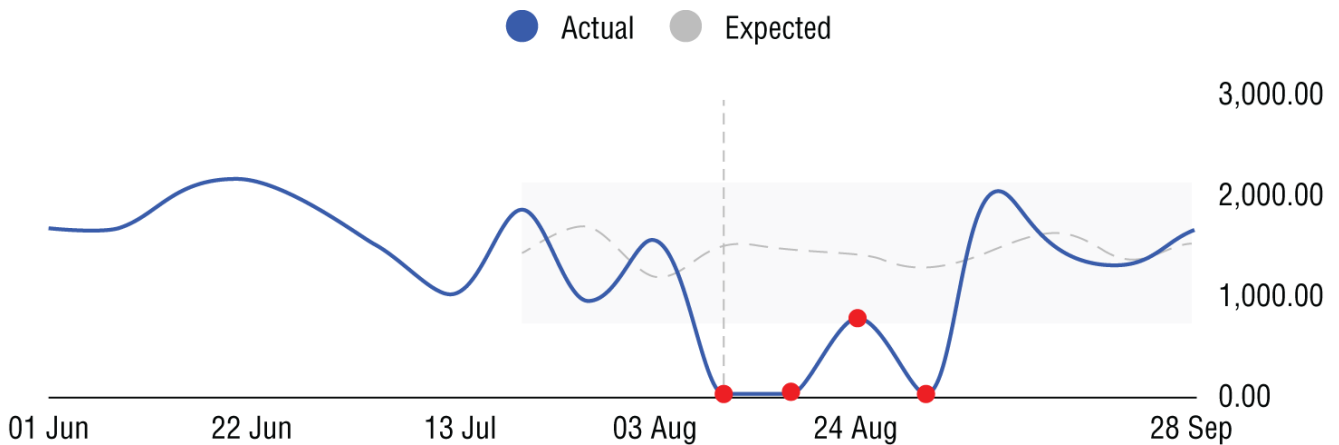


圖14.2 點異常範例

Figure 14.2 Example of point anomalies

來源: 作者 / 經 millimetric.ai 許可轉載 / <https://millimetric.ai/blog/what-is-an-anomaly>

Source: author / reproduced with permission of millimetric.ai / <https://millimetric.ai/blog/what-is-an-anomaly>

靜態閾值顧名思義：系統或用戶可以設置靜態閾值（如 90%）。超過該閾值時，表示異常。在圖 14.3 中，請注意閾值設置得太低（僅 10%），因此藍線不斷誤入紅色“異常區域”。（你必須相信我的顏色，因為這是一本黑白書。這樣的檢測監視器會非常嘈雜，並且大多數時候處於“異常狀態”。（那會很奇怪，因為異常就是常態.....但這似乎就是世界即將走向的樣子。

Static thresholds are exactly as they sound: The system or user can set a static threshold (like 90%). That threshold, when exceeded, signals an anomaly. In Figure 14.3, note that the threshold is set too low (at only 10 percent), so the blue line is constantly straying into the red “ anomaly territory. ” (You ’ ll have to trust me on the colors, since this is a black-and-white book.) Such a detection monitor would be very noisy and in an “ anomaly state ” most of the time. (That would be weird because an anomaly

would be the normality...But that seems to be what the world is coming to.)

正如您從 Datadog 的這個範例中看到的，不涉及太多 AI 機制，而且演算法非常簡單：

As you can see from this example from Datadog, not much AI machinery is involved, and the algorithm is pretty straightforward:

If the value is $>$ threshold, then the value is an anomaly.

另一方面，動態閾值明顯更有趣。用於檢測異常的經典動態閾值方法使用布林帶。這個想法很簡單：首先，我們計算過去 N 天的簡單移動平均線（SMA）；然後，我們確定 SMA 的標準差並將其加倍。SMA + 2 標準差線形成上軌，而 SMA - 2 標準差線形成下軌（見圖 14.4）。

Dynamic thresholds, on the other hand, are significantly more interesting. The classic dynamic threshold method for detecting anomalies uses Bollinger Bands. The idea is simple: First, we calculate a simple moving average (SMA) for the last N days; then, we determine a standard deviation from SMA and double it. The SMA + 2 standard deviations line forms the upper band, while the SMA - 2 standard deviations line forms the lower band (see Figure 14.4).

布林帶對於確定價格異常非常有用。日內交易者發誓，每當股價“突破”該區間時，都是因為股價正在發生重大事件。當這種情況發生時，這是一個異常情況，表明人們應該買入或賣出股票。（這似乎比用雞內臟預測股價要好。嗯，也許吧。無論如何，我們在上一章中介紹了預測。

Bollinger Bands are very useful for determining price anomalies. Day traders swear that whenever a stock price “pushes through” the band, it is because something significant is happening to the stock price. When that happens, it’s an anomaly indicating that one should buy or sell the stock. (This seems better than forecasting the stock price using chicken entrails. Well, maybe. Anyhow, we covered forecasting in the previous chapter.)

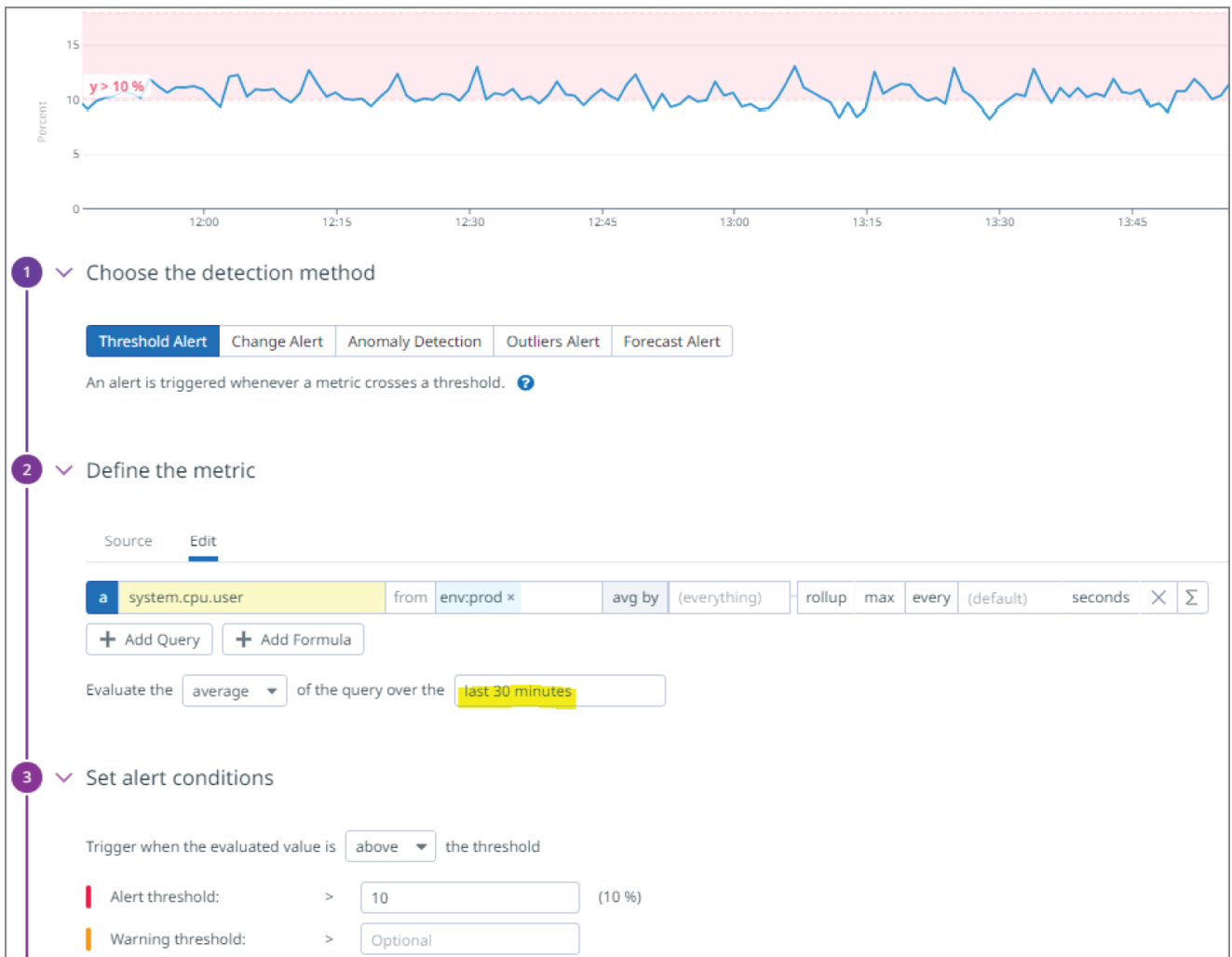


圖 14.3 靜態閾值點異常偵測的 UI

Figure 14.3 The UI for a static threshold point anomaly detection

資料來源：Stack Exchange Inc /

[https://stackoverflow.com/questions/76236544/datadog-only-alert-after-x-amount-of-time /](https://stackoverflow.com/questions/76236544/datadog-only-alert-after-x-amount-of-time/)

最後訪問日期為 2025 年 2 月 5 日

Source: Stack Exchange Inc /

[https://stackoverflow.com/questions/76236544/datadog-only-alert-after-x-amount-of-time /](https://stackoverflow.com/questions/76236544/datadog-only-alert-after-x-amount-of-time/) last accessed on February 05, 2025

無論您是否同意布林帶在每日股價方面的神奇預測品質，事實仍然是布林帶提供了一個簡單的動態閾值解決方案。它們有助於大幅減少誤報的數量，特別是對於可以快速上升和下降並在一段時間內保持在特定值的變數或沒有預設上限或下限的變數。

Regardless of whether or not you agree with the magical predictive qualities of Bollinger Bands when it comes to daily stock prices, the fact remains that Bollinger Bands provide a simple dynamic threshold solution. They help dramatically decrease the number of false positives, especially for variables that can rise and fall quickly and stay at a certain value for a time or variables without preset upper or lower limits.

布林帶只是簡單的數學，並沒有那麼複雜。正如上一章所討論的，簡單的統計方法往往非常有效，而人工智慧有時是不必要的矯枉過正。在這種情況下，布林帶可以成為點異常的有效動態閾值解決方案。但是，根據您的使用情況，簡單的解決方案可能還不夠。您需要複雜的 AI 模型來偵測複雜測量中的異常，並協助減少誤報或誤報。

Bollinger Bands are just simple math and are not that sophisticated. As discussed in the previous chapter, simple statistical methods are often very effective, and AI is sometimes unnecessary overkill. In cases like that, Bollinger Bands can be an effective dynamic threshold solution for point anomalies. However, depending on your use, simple solutions may not be enough. You need sophisticated AI models to detect anomalies in complex measurements and help decrease false positives or false negatives.

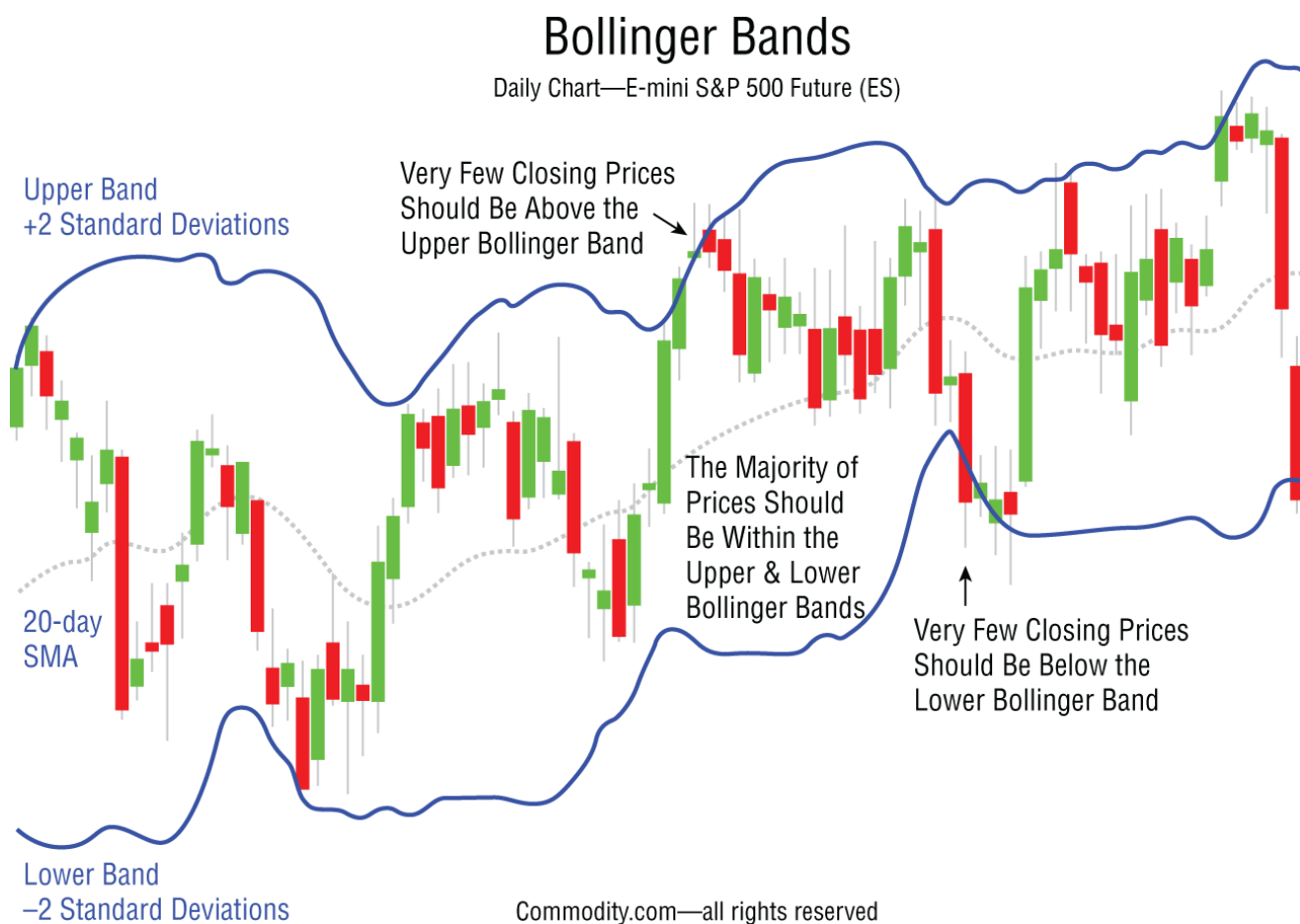


圖 14.4 布林帶是動態閾值點異常檢測的簡單示例

Figure 14.4 Bollinger Bands are a simple example of a dynamic threshold point anomaly detection

來源：作者/經Commodity.com許可轉載

Source: author / reproduced with permission of Commodity.com

圖 14.5 顯示了一個更複雜的 AI 驅動的動態閾值範例（同樣來自 Datadog）。將其與本章前面圖 14.3 所示的靜態閾值圖片進行比較。

Figure 14.5 shows a more complex AI-driven dynamic threshold example (again, from Datadog). Compare it with the static threshold picture shown in Figure 14.3 earlier in the chapter.

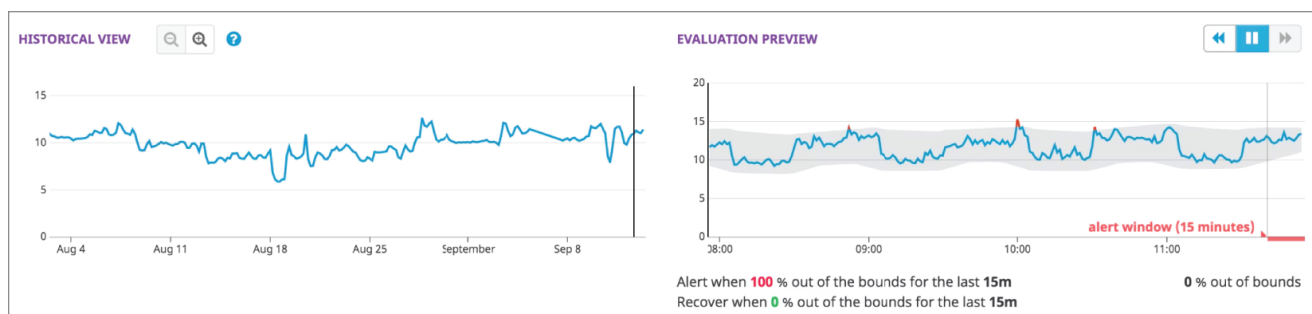


圖 14.5 用於點異常檢測的更複雜的 AI 驅動動態閾值

Figure 14.5 A more complex AI-driven dynamic threshold for point anomaly detection

來源：Datadog / <https://docs.datadoghq.com/monitors/types/anomaly/> / 最後訪問日期：2025 年 2 月 5 日

Source: Datadog / <https://docs.datadoghq.com/monitors/types/anomaly/> / last accessed on February 05, 2025

用於微調 AI 驅動的動態閾值設置的接口可能會變得非常棘手（見圖 14.6）。

Interfaces for fine-tuning AI-driven dynamic threshold settings can get pretty hairy (see Figure 14.6).

讓使用者手動調整數百或數千個指標的此偵測類型是不可行的。這就是為什麼您可能想要設計您的系統，以便 AI 算法可以根據某些學習參數（例如一些誤報和漏報）選擇最佳的自動預設。AI/ML 方法可以幫助優化特定測量的正確標準差（帶）數量（“2”），並以最大限度地減少假陰性和誤報

數量的方式更新 SMA 間隔（“1 週”）。您可能還希望讓 AI 從幾種替代專有檢測算法中進行選擇，例如圖 14.6（2）中名為“敏捷”的算法。

Having users manually adjust this detection type for hundreds or thousands of metrics is just not feasible. That’s why you may want to design your system so that the AI algorithm can select the best automatic presets based on some learning parameters (like a number of false positives and negatives). AI/ML methods can help optimize the correct number of standard deviations (bands) for a particular measurement (“2”) and update the SMA interval (“1 week”) in a way that minimizes the number of false negatives and false positives. You may also wish to have AI select from several alternative proprietary detection algorithms, such as the algorithm named “agile” in Figure 14.6 (2).

The image shows a complex UI for configuring anomaly detection. It is organized into several sections:

- Advanced options:** Includes a "Reset to Auto Defaults" button.
- Anomaly detection algorithm options:** A dropdown menu is set to "agile", followed by "algorithm with" and another dropdown set to "weekly", and "seasonality" with a help icon.
- Timezone:** A toggle switch is turned on, with a dropdown menu set to "America/New York (UTC-04:00)".
- Rollup interval:** A dropdown menu is set to "60" seconds, with a help icon.
- Recovery options:** Includes a dropdown for "Recover if the values are within the bounds for at least" set to "15 minutes". Below are two threshold settings: "Alert recovery threshold" set to "0" (0% of values outside bounds) and "Warning recovery threshold" set to "Optional" (0% - 100%).
- Missing data options:** A dropdown menu is set to "[Never]", with a help icon.
- Evaluation options:** Includes a dropdown for "Delay monitor evaluation by" set to "0" seconds, and another dropdown for "Do not require" a full window of data for evaluation, with a help icon.

圖 14.6 用於微調 AI 驅動的點異常動態閾值設置的 UI 設計可能會變得非常複雜

Figure 14.6 UI design for fine-tuning AI-driven dynamic threshold settings for point anomalies can get pretty complex

來源：Datadog / <https://docs.datadoghq.com/monitors/types/anomaly/> / 最後訪問日期：2025 年 2 月 5 日

Source: Datadog / <https://docs.datadoghq.com/monitors/types/anomaly/> / last accessed on February 05, 2025

敬

WARNING

動態閾值並不適合每次測量。

Dynamic thresholds are not suitable for every measurement.

例如，CPU 忙碌百分比或交易錯誤計數之類的內容可能不是動態臨界值偵測的最佳候選者，因為這些指標的任何非預期尖峰都是嚴重狀況，而且會降低效能。我們不希望系統「學習」隨著時間的推移忽略越來越多的錯誤。其他不適合動態閾值的測量包括合規性、品質控制和服務等級協定（SLA）指標。例如，所需的系統正常運行時間、飲用水中鉛的最大百萬分之一（PPM）或生雞肉在生產嬰兒食品的工業廚房中在室溫下可以放置的最大分鐘數。（毛。我知道。但必須有人追蹤它。既然川普和埃隆的 DOGE 袋解僱了大部分負責消費者保護的聯邦僱員，那麼設計人員可能需要幫助建立一個可用的介面，對關鍵的 FDA 異常指南進行硬編碼，這些指南不應隨著時間的推移而動態變化（3）。

For example, something like CPU Busy Percent or count of transaction errors may not be a great candidate for dynamic threshold detection because any unexpected spike of those metrics is a critical condition and will degrade performance. We don't want the system to "learn" to ignore larger and larger numbers of errors over time. Other measurements unsuitable for dynamic thresholds include compliance, quality control, and service level agreements (SLAs) metrics. Examples are required system uptime, maximum parts per million (PPM) of lead in drinking water, or a maximum number of minutes raw chicken can sit at room temperature in an industrial kitchen that makes baby food. (Gross. I know. But someone's got to track it. Now that Trump and Elon's DOGE-bags fired most of the federal employees in charge of consumer protection, it might be up to designers to help build a usable interface that hard-codes key FDA anomaly guidelines that should not dynamically change over time (3).)

另一方面，網路流量、交易數量、訂單量、會話長度和收入等將是動態閾值監控的完美候選者，因為就像股票價格一樣，這些指標往往會上升或下降，並由於外部因素而保持在特定的動態範圍內一段時間。例如，週末的網路流量可能非常低，而在星期六和星期日調整到較低閾值的系統正是偵測安全漏洞所需要的。然而，到週一早上，我們需要自動重新提高閾

值;否則，由於合法工作活動而產生的網路流量將觸發警報。使用靜態或動態閾值的用例並不總是直觀的，一些用戶體驗研究和深入對話總是一個好主意。

On the other hand, something like network traffic volume, number of transactions, order volume, session length, and revenue, among others, would be perfect candidates for dynamic threshold monitoring because, just like stock prices, those metrics tend to go up or down and stay in a particular dynamic range for some time due to external factors. For example, network traffic volume might be very low during the weekend, and the system that adjusts to a lower threshold during Saturday and Sunday would be just what is needed to detect a security breach. However, by Monday morning, we need to automatically re-up the threshold; otherwise, network traffic due to legitimate work activity would set off the alarms. The use case for using static or dynamic threshold is not always intuitive, and some UX research and in-depth conversations are always a great idea.

便條

NOTE

這就是為什麼身為使用者體驗專業人員，您需要與工程師、資料科學家、客戶和主題專家（SME）面談，以確定哪個閾值適用於您的特定點異常偵測使用案例。

This is why you, as a UX professional, will need to interview engineers, data scientists, customers, and subject matter experts (SMEs) to determine which threshold applies to your particular point anomaly detection use case.

最後，值得一提的是，點異常本身通常並不重要。相反地，特定時段（例如一分鐘或一小時）內的點異常數目要重要得多。偶爾的 CPU 忙碌百分比尖峰不會對效能產生太大影響，但同一分鐘內的多個尖峰會迅速降低機器的回應能力和效能。因此，大多數監控系統都清楚地區分了異常和警報。同樣，通過與用戶、中小企業和數據科學家進行高質量對話進行深入研究始終是一個好主意！

Lastly, it is important to mention that a point anomaly is often not significant in and of itself. Rather, the number of point anomalies within a specific time period (like a minute or an hour) is much more significant. An occasional CPU Busy Percent spike is not that impactful to performance, but multiple spikes within the same minute will rapidly degrade the responsiveness and performance of the machine. Thus, most monitoring systems differentiate clearly between an anomaly and an alert. Again, in-depth research via quality conversations with users, SMEs, and data scientists is always an excellent idea!

變更點異常

Change Point Anomaly

變更點異常在大部分方面與點異常類似。兩者都可以被視為「振幅」異常，例如，每當觀察值意外突破某個靜態或動態閾值時，就會發生異常。變化點和點異常之間的最大區別在於，點異常是迅速回落到基線的“尖峰”，而變化點異常是隨著時間的推移而持續的意外變化。因此，對於每種類型的異常，您調整系統的 UI 方法會略有不同。

Change point anomalies are similar to point anomalies in most respects. Both can be considered “amplitude” anomalies—for example, they occur whenever the observed value unexpectedly breaches a certain static or dynamic threshold. The big difference between change point and point anomalies is that point anomalies are “spikes” that quickly come back down to baseline, and a change point anomaly is an unexpected change that remains sustained over time. Therefore, your approach to UI for tuning the system will be slightly different for each type of anomaly.

當您建構介面以調整點異常偵測並判斷是否需要警示時，您想要提供一種方法，讓使用者指出點異常在特定時段內發生的次數。例如，如果一分鐘內有三個或更多異常的 CPU 忙碌百分比尖峰，系統會觸發警示（請參閱圖 14.7）。

When you construct the interface to tune point anomaly detection and determine whether an alert is needed, you want to provide a way for users to indicate the number of times the point anomaly occurred within a certain period of time. For example, if there are three or more anomalous CPU Busy Percent spikes in one minute, the system triggers an alert (see Figure 14.7).

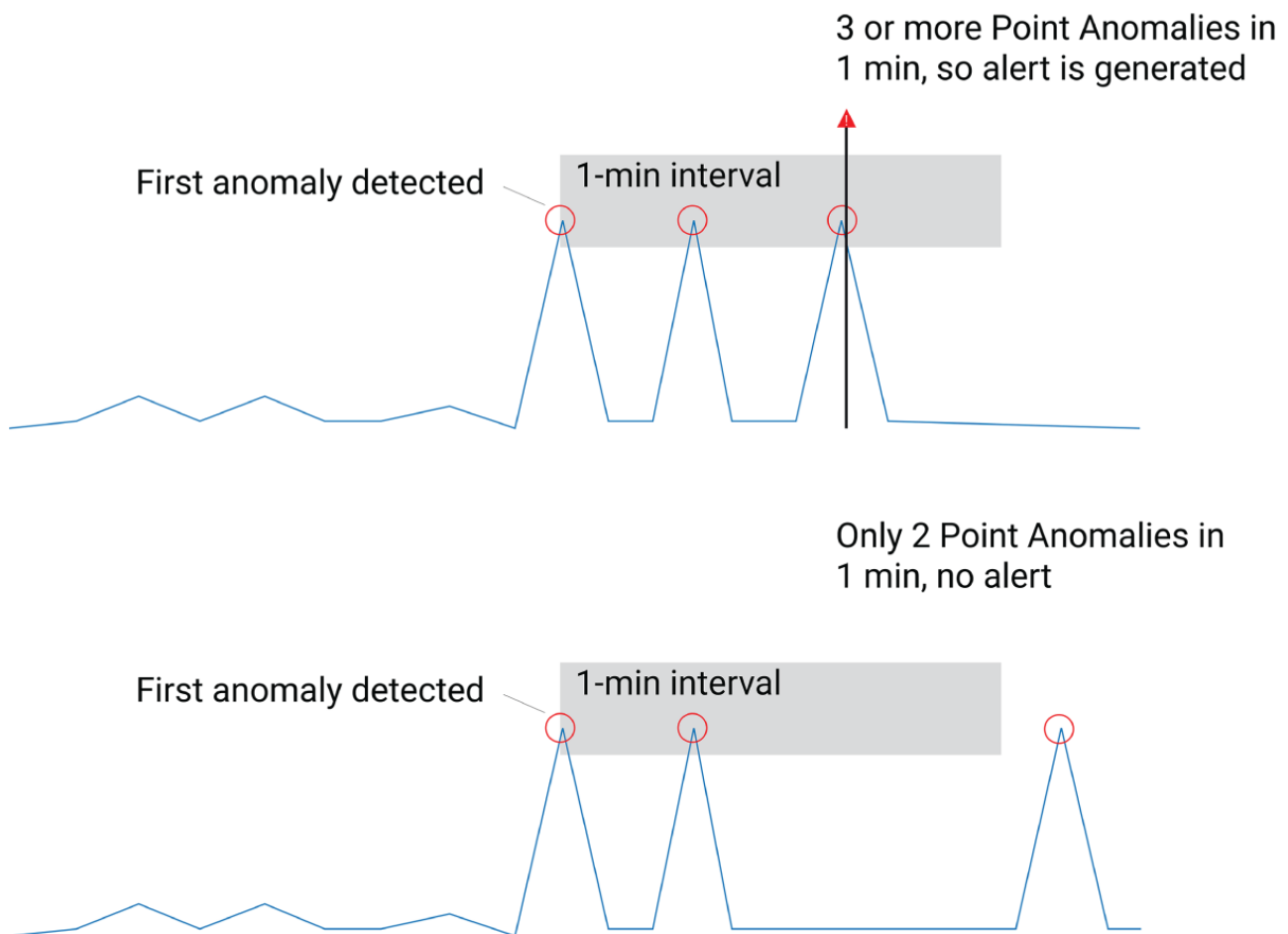


圖 14.7 只有在一分鐘間隔內偵測到三個或更多異常時，才會產生點異常偵測警示

Figure 14.7 A point anomaly detection alert is only generated if three or more anomalies are detected in a one-minute interval

相反地，當您建構介面來調整變更點異常偵測以查看是否需要產生警示時，您會想要介面指出異常讀數持續異常值的時間長度；例如，如果該值在超過一分鐘的時間內大於某個閾值，則系統會觸發警報（見圖 14.8）。

In contrast, when you construct the interface to tune change point anomaly detection to see if it needs to generate an alert, you want your interface to indicate how long the anomalous reading sustained an abnormal value; for example, if the value was greater than a certain threshold for over one minute, then the system triggers the alert (see Figure 14.8).

區分變更點和點異常可協助您針對特定使用案例微調系統，但在許多使用案例中，您不需要進行此區分。通常會看到允許用戶在相同配置表單中處理這兩個條件的 UI。使用速記符號（[值 V] = 代表下拉式選單，[值] = 文字欄位，[X] = 複選框等 - 稍後會詳細介紹），我們可以將其簡單地寫成

Differentiating between change point and point anomalies helps you fine-tune the system for the specific use case, but in many use cases, you will not need to make this distinction. It is common to see the UI that allows the user to take care of both conditions in the same configuration form. Using a shorthand notation ([value ∨] = stands for a drop-down, [value] = text field, and [X] = checkbox, etc.—more on this later), we can write it down simply as

Trigger a [Critical ∨] alert whenever:

[X] the value exceeds [90 %] for a period of [1 minute ∨]

[X] [3 times ∨] in a [1 minute ∨] interval

因此，您可以使用單一 UI

涵蓋這兩種類型的異常。動態和靜態閾值對於這兩種類型的異常應該遵循類似的原則。

Thus, you are covering both types of anomalies with a single UI. Dynamic and static thresholds should follow similar principles for both types of anomalies.

如果系統使用 AI/ML 方法自動微調這些值，並且僅通過將警報指定為“真陽性”或“誤報”來“要求”用戶響應警報，以便它可以學習和改進算法以獲得最佳性能，這是理想的選擇。然而，在所有情況下，提供一種手動方法來覆蓋 AI 值是絕對必要的（有關維持人類對 AI 監督的重要性的更多信息，請參閱本書的第 4 部分）。

It is ideal if the system fine-tunes these values automatically using AI/ML methods and only “asks” the user to respond to alerts by designating them as “true positive” or “false positive” so it can learn and improve the algorithm for best performance. However, in all cases, providing a manual way to override the AI values is absolutely essential (see Part 4 of this book for more on the criticality of maintaining human oversight over AI).

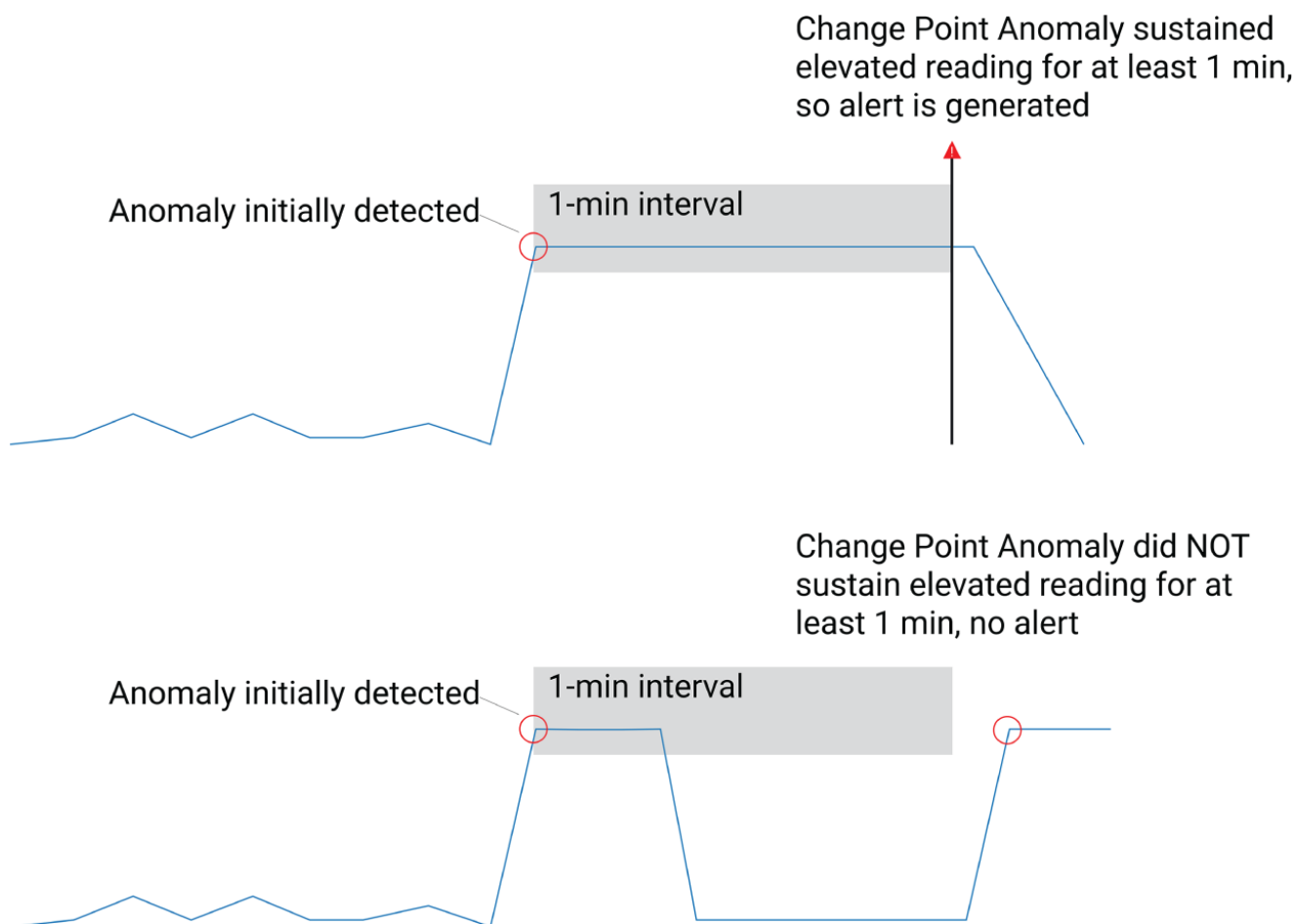


圖 14.8 只有在異常持續升高讀數一分鐘或更長時間時，才會產生變更點異常偵測警示

Figure 14.8 A change point anomaly detection alert is only generated if the anomaly sustained an elevated reading for one minute or longer

以下是設計點和變更點異常偵測 UI 的五個關鍵原則：

Here are the five key principles for designing the point and change point anomaly detection UIs:

- 根據您的使用案例，如果異常發生一次，則可能不會引起警報，如果我們提醒系統使用者每個異常，則會出現太多誤報。若要規範警示數目，您的 UI 通常需要具有某種「發生計時器」，無論是針對發生次數（針對點異常）或變更持續時間（針對變更點異常）。 Depending on your use case, if an anomaly occurs once, it may not be a cause for alarm— if we alert the system 's users about every anomaly, there will be too many false positives. To regulate the number of alerts, your UI often needs to have some kind of “ occurrence timer, ” either for a count of occurrences (for point anomalies) or for the duration of the change (for change point anomalies).

- 優先提供折線圖（過去 15 分鐘/1 小時/24 小時等）作為模型，以允許用戶實時預覽他們的更改將對相當大小的歷史數據樣本產生的影響。這將幫助用戶避免無意中產生過多的誤報或遺漏感興趣的訊號（並產生誤報）。Make it a priority to provide a line graph (for the past 15 minutes/1 hour/24 hours, etc.) as a model to allow users to preview the effects their changes will have on a decent-sized sample of historical data in real time. This will help users avoid inadvertently creating too many false positives or missing signals of interest (and creating false negatives).
- 對於複雜的動態演算法調優，請嘗試使用 Mad Lib 的「填空」UI 設計模式，讓整個調優 UI 讀起來像英文句子。內嵌在文字中的下拉式功能表和文字欄位提供輸入。這將有助於即使不熟悉該主題的用戶也能了解調整參數在其他設置中的作用。For complex dynamic algorithm tuning, try using a Mad Lib “fill-in-the-blank” UI design pattern so that the entire tuning UI reads like an English sentence. Drop-down menus and text fields embedded in the text provide the inputs. This will help even users unfamiliar with the subject understand what the tuning parameters do in the context of other settings.
- 使用 AI/ML 方法建立合理的預設值來微調 UI，以最大限度地減少使用者的工作量。學習演算法也可用於自我調整系統，特別是在需要持續監控數千個物件的數百個變數的情況下。Use AI/ML methods to create reasonable defaults for fine-tuning the UI to minimize user effort. Learning algorithms can also be used to self-tune the system, especially in cases where hundreds of variables for thousands of objects need to be monitored on an ongoing basis.
- 切勿停用手動調整方法，特別是在 AI/ML 執行微調的情況下。Never disable manual tuning methods, especially for cases where the AI/ML is performing the fine-tuning.

上下文異常

Contextual Anomaly

點和變化點異常是當感興趣的變量的值突破某些靜態或動態數值閾值時形成的“振幅”異常。相比之下，更複雜和有趣的上下文異常屬於“隨時間變化的形狀變化”異常系列——例如，我們不是分析變量的值，而是分析線條隨時間形成的形狀，並據此確定變量的行為是否異常。有時，有一種方法可以用數學方式描述複雜的週期性出現的形狀，但形狀異常檢測通常使用 AI/ML 方法完成，這通常非常適合任務（4）。

Point and change point anomalies are “ amplitude ” anomalies formed when a value of a variable of interest breaches some static or dynamic numerical threshold. In contrast, the more complex and interesting contextual anomalies fall into a family of “ shape change over time ” anomalies—for example, rather than analyzing the value of a variable, we analyze the shape the line makes over time and determine if the behavior of a variable is anomalous based on that. Sometimes, there is a way to describe complex periodically occurring shapes mathematically but shape anomaly detection is often done with AI/ML methods, which typically work quite well for the task (4).

雖然上下文異常有數千個用例，但作者更熟悉以下幾個用例：

While there are many thousands of use cases for contextual anomalies, here are a few that the authors are familiar with more intimately:

Unseasonable Traffic Spike: Suppose a website receives the expected amount of traffic during working hours Monday through Friday and suddenly experiences too little or too much traffic that does not match the expected behavior for a particular time of day or day of the week. That is a contextual anomaly related to hourly or weekly seasonality. Contextual anomalies of this kind can signal security breaches (traffic during unexpected times, unexpected volume of traffic from unexpected countries/IP addresses, unexpected locations of admin traffic, etc.).

Unseasonable Traffic Drop: If your Add To Cart button is broken and the expected sales suddenly drop, or a poorly configured discount code reduces the price to \$0, the revenue drops precipitously. These bugs have a tangible impact on the business and can be quite costly. Contextual anomaly monitoring can help detect these bugs and fix them quickly.

Slow Drift: Sometimes, the trend is not a sudden change but a slow, gradual drift over time. These types of anomalies are devilishly difficult to detect. For example, if your online traffic keeps increasing over time, it ’ s not a bad thing. Still, it is an anomaly that might warrant an action, like increasing the computing capacity of your web server.

Machine Vibration Anomalies: When a complex machine (engine, pump, turbine, airplane wing, factory equipment, etc.) is operating, it vibrates a certain way. That vibration can be measured using a dynamometer (a device that measures force) over time. A trace of vibration over time has very distinctive peaks and troughs, which can be used to detect deviation from a “ healthy ” or expected “ shape ” of vibration. For example, anyone who owns a car and a functional eardrum can tell you that a cold car that just started up on a freezing sub-zero-degree morning will vibrate quite differently from a

warmed-up vehicle. Likewise, an astute car owner can usually detect weird noises, knocking, etc., which result from a change in a pattern and frequency of vibration and can be used to detect anomalies and predict if a piece of equipment is about to break down. In some cases, contextual vibration anomalies can even be used to predict the nature of the breakdown and estimate the time to failure.

便條

NOTE

請記住，雖然您的產業可能與此處所述的產業不同，但瞭解此類異常的 UI 最佳實務將使您能夠發現適用於特定產品或產業的現有（或發明新穎）形狀異常使用案例。

Keep in mind that while your industry might be different than those described here, learning the UI best practices for these kinds of anomalies will enable you to discover existing (or invent novel) shape anomaly use cases that apply to your specific product or industry.

基於季節性的上下文異常

Contextual Anomaly Based on Seasonality

以下是 Jepto 季節性形狀異常偵測 UI 的絕佳範例。這是一個乾淨、直接的介面，用於配置異常演算法（見圖 14.9）。

The following is an excellent example of seasonal shape anomaly detection UI from Jepto. It 's a clean and straightforward interface for configuring the anomaly algorithm (see Figure 14.9).

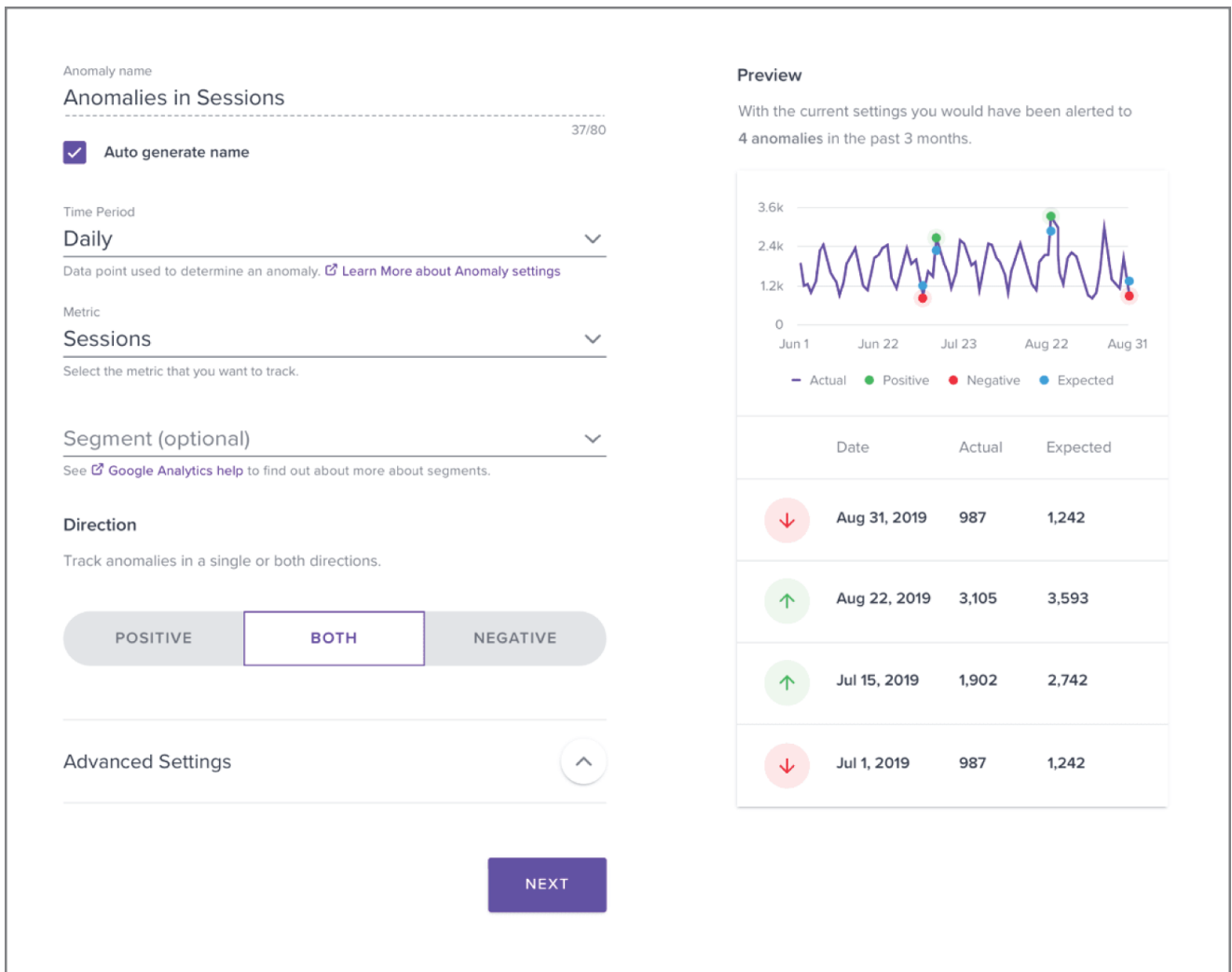


圖 14.9 季節性形狀異常偵測 UI

Figure 14.9 Seasonal shape anomaly detection UI

資料來源：Jepto / <https://www.jepto.com/help/anomaly-detection-settings/>

最後訪問日期：2025 年 2 月 5 日

Source: Jepto / <https://www.jepto.com/help/anomaly-detection-settings/> / last accessed on February 05, 2025

主要的兩個可配置參數是

The main two configurable parameters are

- 時間段（每日）：決定演算法將嘗試建模的週期時間範圍 Time Period (Daily): Determines the time frame of periodicity that the algorithm will try to model

- 方向（兩者）：如果值超過預期或低於預期，是否會產生異常 Direction (Both): Whether the anomaly will be generated if the value surpasses the expectation or underwhelms it

這兩個參數都應該簡單易懂，使用者可以理解。

Both of these parameters should be straightforward for users to understand.

另請注意我們之前提到的另一個功能：過去時間段的預覽圖，顯示如果這些設置生效，過去的數據會產生多少異常。此預覽窗格很方便，因為它可以避免錯誤地設定系統並產生太多誤報（被誤解為異常的正常讀數）或誤報（應該發生異常但未報告）。

Also note another feature we mentioned earlier: the preview graph of the past time periods that shows how many anomalies would have been generated from the past data if these settings had been in effect. This preview pane is handy because it avoids setting up the system incorrectly and creating too many false positives (normal readings that are misinterpreted as anomalies) or false negatives (where an anomaly should have happened but was not reported).

大多數有趣的設定都隱藏在「進階」窗格中（見圖 14.10），因為大多數時候，大多數使用者會相信基本演算法設定會做得很好。

Most of the interesting settings are hidden in the Advanced pane (see Figure 14.10) because, most of the time, the majority of users will trust the basic algorithm settings to do a good job.

最重要的進階設定是正異常閾值。它決定了可以指定為異常的資料百分比。此指標調整系統的靈敏度。正如設計界的諺語所說，“如果一切都是大膽的，那麼就沒有什麼是大膽的。這同樣適用於異常。（此指標也與處理能力有關：一個團隊可以在合理的時間內查看多少異常情況？請參閱側邊欄“可變 AI 模型的成本和收益”，第 5 章“價值矩陣——AI 準確性是胡說八道。這就是使用者體驗必須對此採取的措施。

The most important advanced setting is the Threshold of Positive Anomalies. It determines what percentage of the data can be specified as anomalies. This metric adjusts the system's sensitivity. As the saying in design goes, “If everything is bold, nothing is.” The same applies to anomalies. (This metric is also related to processing capacity: how many anomalies can a team look at in a reasonable amount of time? See the sidebar “The Costs and Benefits in Variable AI Models” in Chapter 5, “Value Matrix—AI Accuracy Is Bullshit. Here's What UX Must Do About It.”)

改進的機會

Opportunities for Improvement

雖然大多數人都會同意這個 UI 乾淨、執行良好、簡單，並且提供了大量的上下文幫助，但表單操作的概念相當複雜。在許多領域，可以使用從 UX 中收集的最佳實踐來改進此 UI，以進行 AI 人為因素研究。雖然您的里程可能有所不同，但這裡有一個簡短的問題列表，以及建議的解決方案，以嘗試用於您的下一個 UX for AI 異常檢測項目。

While most folks would agree that this UI is clean, well-executed, straightforward, and provides a good amount of contextual help, the concepts the form manipulates are fairly sophisticated. There are many areas where this UI can be improved using best practices that can be gleaned from UX for AI human factors studies. While your mileage may differ, here ' s a short list of problems, along with suggested solutions to try for your next UX for AI anomaly detection project.

Only a Single Periodicity can be Chosen: In the current UI, the user must pick the period: daily, weekly, monthly, yearly, etc. However, we already discussed that most periodicity happens as a combination of these: during the day, there is low demand at 2 – 6 a.m., for instance (daily pattern), then the typical decreased demand on weekends (weekly pattern). Finally, during holidays in November/December, the traffic spikes (yearly pattern). Picking only a single periodicity, such as “ weekly, ” misses the opportunity to model these more complex cycles and identify important anomalies (such as a traffic spike at 3 a.m.) while generating many false positives during the holidays when the traffic is high. Rather than having the user pick one periodicity, the ideal UI should automatically suggest an anomaly schedule that best fits the available data (likely a combination of daily, weekly, and yearly trends).

The Algorithm is not Self-balancing: This is important because during certain times, like seasonal high shopping demand, the data will be noisy and out of balance, generating many false positives. While there are settings in the UI to help ameliorate this, they need to be understood and then set manually. To remedy the problem, consider having a setting that automatically adjusts based on an unusually high number of anomalies. If this has to be user-driven, it can be placed closer to the source of the problem, such as a simple question that is triggered from the alert, for example (using shorthand notation, where [] = checkbox):

[] Getting too many alerts on this metric? Check this box to adjust anomaly levels automatically based on recent data.

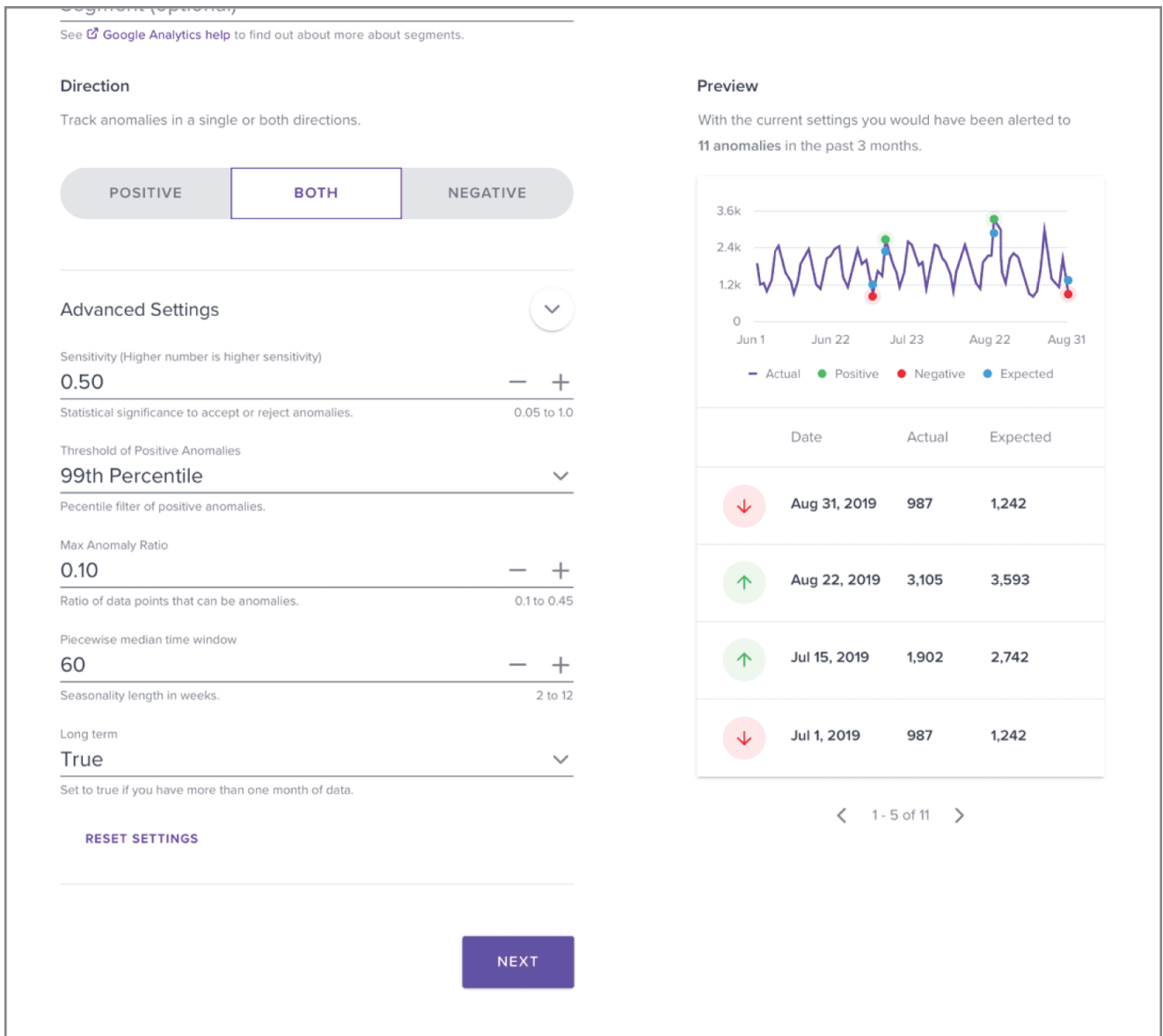


圖 14.10 季節性形狀異常偵測 UI 的進階設定

Figure 14.10 Advanced settings for seasonal shape anomaly detection UI

資料來源：Jepto / <https://www.jepto.com/help/anomaly-detection-settings/>

最後訪問日期：2025 年 2 月 5 日

Source: Jepto / <https://www.jepto.com/help/anomaly-detection-settings/> / last accessed on February 05, 2025

The Sensitivity Drop-down Menu is Hidden: As discussed earlier, the most critical system setting is the Threshold of Positive Anomalies. Unfortunately, it is hidden under the Advanced settings, perhaps to protect the user from the hard-to-understand values in the drop-down list.

The Sensitivity Options in the Drop-down Menu are Confusing: The options in the sensitivity drop-down menu are confusing. According to the documentation, they are: 99th Percentile, 95th Percentile, Median of The Data Max Values, and Default Setting for Digital Marketing Data. I suspect most folks will have trouble understanding these options. Instead, consider having a simple slider like this one (again using shorthand notation):

Number of Anomalies -----\----- [X] adjust automatically

預設情況下，會選取 [自動] 選項，以符合資料的採用設定，並提供滑桿，以防使用者想要手動調整異常數目。像這樣的滑桿比百分位數下拉式選單直觀得多：向右拖曳控點可偵測更多異常，向左拖曳可降低靈敏度並偵測較少異常。

The Auto option is selected by default to the adoptive setting that fits the data, and the slider is provided in case the user wants to manually adjust the number of anomalies. A slider like this is much more intuitive than a percentile drop-down menu: Drag the handle to the right to detect more anomalies or to the left to decrease sensitivity and detect fewer anomalies.

Add the Anomalies Recommendations in the Copilot: In addition to direct settings in the UI, a Copilot chat functionality can be used to provide user guidance and help choose the correct settings. Copilot can help users adjust settings recommendations based on historical data, answer specific documentation questions, play out “ what if ” scenarios, and more. (See Chapter 12, “ Modern Information Architecture for AI-First Applications, ” for more discussion of AI-first settings pages.)

最重要的要點值得重複：正如我們在上一章中討論的，電子商務網站的典型週期是每小時（夜間流量較少）、每週（週六和周日流量較少）和每年（假期前流量增加）和典型銷售時間的組合。因此，AI/ML 工具將比典型演算法做得更好，因為它們基於使用先前的歷史資料而不是演算法預測直接訓練模型。識別週期性的正確組合，然後使用它來準確識別上下文異常，可以成為人工智慧技術的一個很好的應用。

The most important takeaway bears repeating: As we discussed in the previous chapter, typical periodicity for an e-commerce site is a combination of hourly (less traffic at night), weekly (less traffic on Saturday and Sunday), and yearly (more traffic right before the holidays and typical sales times). Thus, AI/ML tools will do much better than typical algorithms because they are based on direct training of models using previous historical data and not algorithmic predictions. Identifying the correct combination of periodicity and then using it to identify contextual anomalies accurately can be an excellent application of AI technology.

雖然手動的單一設定 UI 似乎適合設定一兩個異常，但它的擴展性不佳。回想一下，這些設定可能需要在數百個（如果不是數千個）指標中以不同的方式應用。自動化和自我平衡可能是關鍵。仔細考慮使用案例，並根據您的任務規模和客戶的教育程度制定適合您的使用者體驗設計建議。

While a manual, single-setting UI seems appropriate for configuring one or two anomalies, it doesn't scale well. Recall that these settings will likely need to be applied differently across hundreds, if not thousands, of metrics. Automation and self-balancing will likely be the key. Carefully consider use cases and make your UX design recommendations appropriate to the scale of the task and your customers' levels of education.

曲線形狀異常

Curve Shape Anomalies

到目前為止，我們已經討論了振幅（點異常和變化點異常）和形狀隨時間的變化（上下文異常）。在最後一個範例中，我們將檢查曲線形狀異常。檢測曲線形狀異常不是查看值的變化或週期性的中斷，而是純粹依賴於人工智慧對曲線形狀有時逐漸變化的識別。

So far, we've discussed amplitude (point anomaly and change point anomaly) and shape change over time (contextual anomaly). In our final example, we will examine the curve shape anomaly. Rather than looking at a change in value or break in periodicity, detecting curve shape anomalies relies purely on AI's recognition of the sometimes gradual change in the curve's shape.

我們將使用馬頭油泵作為本節的示例用例。您將看到，可以訓練 AI 識別由移動組件描述的各种曲線形狀，以確定系統的狀態並授權糾正措施。

We'll use the Horse-Head oil pump as an example use case for this section. You'll see that AI can be trained to recognize various curve shapes described by moving components to determine the system's state and authorize corrective action.

馬頭油泵

Horse-Head Oil Pump

最常用的油泵類型之一是馬頭或梁式油泵千斤頂，由 Walter Trout 於 1952 年發明。然而，最初的設計要古老得多，可以追溯到中國古代的灌溉系統（5）。電動機旋轉變速箱，振盪橫梁，使馬頭（和連接的桿）上下移動，使泵將油帶到表面。圖 14.11 顯示了完整的數位孿生（請參閱第 4 章「數位孿生 - 系統實體元件的數位表示」）。

One of the most commonly used types of oil pumps is the Horse-Head, or Beam, oil pumpjack, invented by Walter Trout in 1952. However, the original design is much older, harking back to the ancient Chinese irrigation systems (5). The electric motor spins the gearbox, oscillating the beam, making the Horse-Head (and the attached rod) move up and down, allowing the pump to bring oil to the surface. Figure 14.11 shows the complete digital twin (see Chapter 4, “ Digital Twin—Digital Representation of the Physical Components of Your System ”).

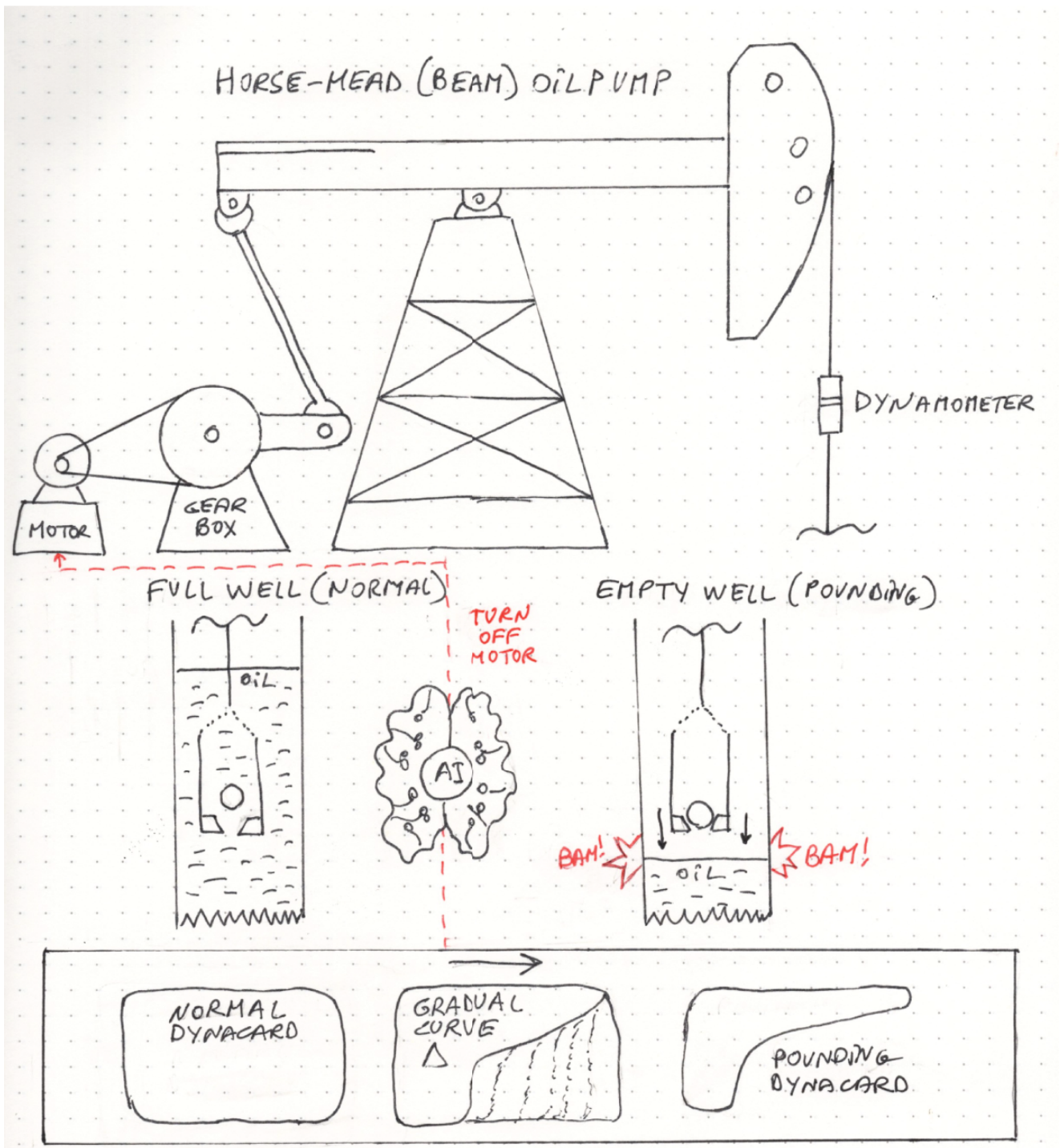


圖 14.11 馬頭油泵的數位孿生。請注意圖紙底部的井下發電機卡

Figure 14.11 Digital twin of a Horse-Head oil pump. Note the downhole dynacards at the bottom of the drawing

我們對這個系統最感興趣的是井深處或“井下”發生的情況，當我們將數噸粘性物質帶到地表時，在動態環境中的地下深處可能會出現許多不同的問題。為了簡潔起見，讓我們只看一種常見的異常情況，稱為“重擊”。

What interests us most about this system is what happens deep in the well, or “ downhole, ” where many different problems might arise deep in the ground in a dynamic environment while we are bringing several tons of viscous material to the surface. For the sake of brevity, let ’ s look at just one common anomaly condition called “ pounding. ”

想像一下，一口傳統的水井，水桶系在一根長繩上。在左邊的井下圖片中，井已經滿了，所以水桶一直坐在水中。上下移動桶很順暢，因為桶永遠不會破壞液體表面。相比之下，右圖顯示的是一口幾乎空井。這張圖中的水桶每次上下移動時都會猛地撞擊在表面上。油比水粘稠得多，因此用鏟斗“ 敲擊 ” 油表面不僅沒有生產力（因為它無助於將油移動到表面），而且會很快導致精密的桿和鏟斗系統彎曲和斷裂！

Imagine, for a moment, a traditional water well with a bucket attached to a long rope. In the downhole picture on the left, the well is full, so the bucket sits all the way in the water. Moving the bucket up and down is smooth because the bucket never breaks the surface of the liquid. In contrast, the picture on the right shows an almost empty well. The bucket in this picture slams down on the surface every time it is moved up and down. Oil is much more viscous than water, so “ pounding ” the oil surface with a bucket is not only nonproductive (as it does not help move the oil to the surface) but will very quickly cause the delicate rod and bucket system to bend and break!

許多快速抽水的油井偶爾會耗盡石油，需要時間再次填滿以防止撞擊。出於這個原因，我們需要弄清楚「 井下 」發生了什麼——這是一種檢測正在發生撞擊的可靠方法，這樣我們就可以快速停止泵浦並等待井再次裝滿。我們還需要一個系統來自主地做到這一點，因為油井通常分佈在很遠的惡劣環境中（例如伊拉克油田，或者咳咳，加利福尼亞州貝克斯菲爾德）。

Many wells pumped at a fast pace occasionally run out of oil and need time to fill up again to prevent pounding. For this reason, we need to figure out what is happening “ downhole ” —a reliable way to detect that pounding is occurring so we can stop the pump quickly and wait for the well to fill up again. We also need a system to do this autonomously because oil wells are often spread out over a great distance in hostile environments (like the Iraqi oil fields or, ahem, Bakersfield, CA.)

幸運的是，我們有一個出色的工具來完成這項工作：我們可以使用連接到桿上的測力計測量桿上的力，然後（使用一些花哨的數學）生成一個漂亮的圖表，稱為“ 井下發電機 ”，其形狀可用於診斷許多不同的情況，包括“ 撞擊 ”。一些示例發電機圖顯示在圖 14.11 中數字學生圖的底部。

Fortunately, we have an excellent tool for the job: We can measure the force on the rod using a dynamometer attached to the rod and then (with a bit of fancy math) produce a nice graph called the “ Downhole Dynacard ” the shape of which can be used to diagnose many different conditions including “ pounding. ” Some sample dynacard graphs are shown at the bottom of our digital twin diagram in Figure 14.11.

通過簡單的目視檢查，您可以看到右側“重擊”發電機的“凹陷”外觀與正常的圓角“方形”形狀有很大不同（如左圖所示）。可以訓練人工智慧識別各種曲線形狀並採取適當的行動（例如關閉泵送馬達並等待設定的時間間隔讓油井再次充滿油）。

By simple visual inspection, you can see that the “ dented ” look of the “ pounding ” dynacard on the right is quite different from the normal “ square ” shape with rounded corners (pictured on the left). AI can be trained to recognize various curve shapes and take appropriate actions (like turning off the pumping motor and waiting for a set interval for the well to fill up with oil once more).

雖然對所有可能的曲線形狀的完整分析超出了本書的範圍，但我們應該提到，一個相對簡單的 AI/ML 系統可以訓練來識別數百種形狀並診斷數十種不同的泵浦條件（6）。

While the complete analysis of all of the possible curve shapes exceeds the scope of the book, we should mention that a relatively simple AI/ML system can be trained to recognize hundreds of shapes and diagnose dozens of different pump conditions (6).

我們需要什麼樣的 UI 來控制這個系統？

What Kind of UI Do We Need to Control This System?

有趣的是，雖然檢測各種條件的機制相當複雜，但 UI 本身並沒有太多功能。請注意，曲線不會立即改變形狀，而是會逐漸改變。隨著每個泵循環，曲線會略有變化，直到……砰！我們受到了重擊。隨著時間的推移，這種微小的變化對我們有利，因為我們不需要採取激烈的行動——我們只想將泵系統的長期損壞程度降到最低。在每個 N 個泵循環（例如 N = 10）之後，我們可以使用新的測功機讀數來生成井下測功機曲線，該曲線可以輸入到本地 AI “ 大腦 ” 中，以與 AI 訓練過的各種測功機進行比較。如果人工智慧能夠自信地回答，根據曲線的形狀，確實發生了「撞擊」，那麼人工智慧就可以做出停止泵送馬達的自主決定。

Interestingly, while the mechanism of detecting various conditions is fairly sophisticated, there is not much to the UI itself. Note that the curve does not change shape immediately but undergoes a

gradual change. With each pump cycle, the curve changes slightly until ... BAM! We got pounding. This slight change over time works to our advantage because we don't need to take drastic action—we just want to minimize the extent of the long-term damage to the pump system. After each N pump cycles (say N = 10), we can use a fresh dynamometer reading to generate a downhole dynacard curve, which can be fed into the local AI “brain” to compare against a variety of dynacards the AI has been trained on. If the AI can answer with some measure of confidence that, based on the shape of the curve, “pounding” is indeed occurring, the AI can make an autonomous decision to stop the pumping motor.

用戶界面非常簡單。同樣，使用我們的速記 UI 符號（請參閱側邊欄“為 AI-pocalypse 做好準備：速記 UX 設計符號作為 AI 提示”），這是一個包含我們所需一切的“核心”的表單：

The UI is quite simple. Again, using our shorthand UI notation (see the sidebar, “Getting Ready for AI-pocalypse: Shorthand UX Design Notation as AI Prompt”), here's a form that has the “meat” of everything we need:

Perform curve anomaly analysis each [10 V] complete pump revolutions

AI Curve Analysis Rule: [Pounding V]

When the curve match confidence \geq [90% V],

perform the following action: [Turn off the pump motor V]

for the period of [10 minutes V] 就是這樣！

That's it!

設計練習：建立您自己的異常偵測 UI

Design Exercise: Create Your Own Anomaly Detection UI

異常偵測是人工智慧提供的重要附加價值。對於您的設計練習，請確定系統中的哪些指標需要監控異常以及原因。

Anomaly detection is a vital value-add provided by AI. For your design exercise, determine which metrics in your system need to be monitored for anomalies and why.

1. 您應該偵測哪些點異常？靜態閾值適用於哪些用例？動態閾值是否有用例，如果有，它是什麼？可以使用布林帶算法的一些變體嗎？如果是這樣，SMA 將基於哪個時間段，以及需要使用多少個標準差（帶子）？您應該與誰交談才能了解詳細信息？What point anomalies should you be detecting? In what use cases would a static threshold apply? Is there a use case for a dynamic threshold, and if so, what is it? Can some variation of the Bollinger Bands algorithm be used? If so, what time period will the SMA be based on, and how many standard deviations (bands) will need to be utilized? Who should you be talking to find out the details?
2. 您可以檢測到哪些變化點異常（例如，一段時間內的持續變化）？異常應持續多長時間，測量值才會被視為異常？What change point anomaly (e.g., persistent change over some period of time) can you detect? How long should the anomaly persist for the measurement to be considered anomalous?
3. 您可以檢測到哪些上下文（隨時間變化的形狀）異常？檢測這些異常的理想測量的潛在週期（每天、每週、每年）是多少？What contextual (shape-change-over-time) anomalies can you detect? What 's the potential periodicity of the measurement (daily, weekly, yearly) ideal for detecting these anomalies?
4. 您可以檢測到哪些曲線形狀異常（請參閱馬頭泵測功卡示例）？偵測到特定條件時，系統應該執行什麼動作？What curve shape anomalies (see the Horse-Head pump dynacard example) can you detect? What action should the system perform when a specific condition is detected?
5. 除了這裡提到的四種異常之外，您還能想到在您的系統中檢測到哪些類型的異常？What kinds of anomalies other than the four mentioned here can you think of detecting in your system?

如果您需要靈感，請考慮以下範例。在完成自己的設計練習之前，不要繼續下一章。

If you need inspiration, consider the following example. Do not proceed to the next chapter until you have completed your own design exercise.

設計練習範例：生命時鐘異常偵測 UI

Design Exercise Example: Life Clock Anomaly Detection UI

我們的用例 Death/Life Clock 具有檢測各種類型異常的豐富可能性。讓我們按類型分解它們，並給出每個類型的範例：

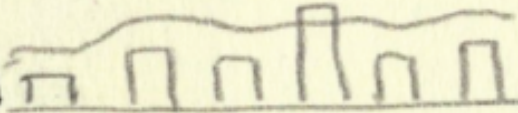
Our use case, Death/Life Clock, is rich with possibilities for detecting various types of anomalies. Let ' s break them down by type, giving an example of each:

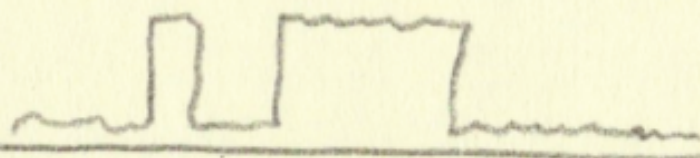
1. 點異常，例如熱量攝取突然激增Point anomaly, such as a sudden spike in caloric intake
2. 變化點異常，例如睡眠期間清醒時間延長Change point anomaly, such as a prolonged wakeful period during sleep
3. 內容異常，例如使用者通常運動的一週中某一天的運動時間意外偏低Contextual anomaly, such as unexpectedly low exercise time during the day of the week when the user typically exercises
4. 曲線形狀異常，例如前後照片中顯示的身體成分的不良變化Curve shape anomaly, such as adverse change in body composition shown in before-and-after photos

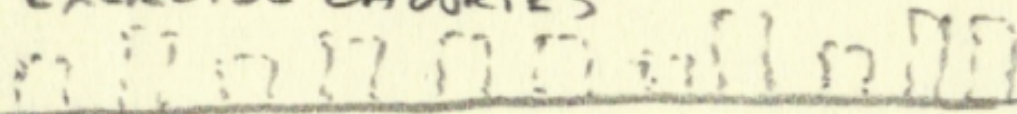
我們可以將所有可能的異常一起列在一個區域中，而不是單獨繪製每個異常，例如名稱缺乏想像力的「異常」列表（見圖 14.12）。

Rather than drawing each anomaly separately, we can list all possible anomalies together in one area, such as the unimaginatively named “ Anomalies ” list (see Figure 14.12).

≡ LIFE COPILOT ANOMALIES

① 
ABNORMAL CALORIC INTAKE
3 DAYS AGO.

② WAKE
SLEEP 
ABNORMALLY LONG WAKE
PERIOD DETECTED LAST
NIGHT.

③ EXERCISE CALORIES

UNEXPECTEDLY LOW AMOUNT
OF EXERCISE DETECTED THIS
WEEK.

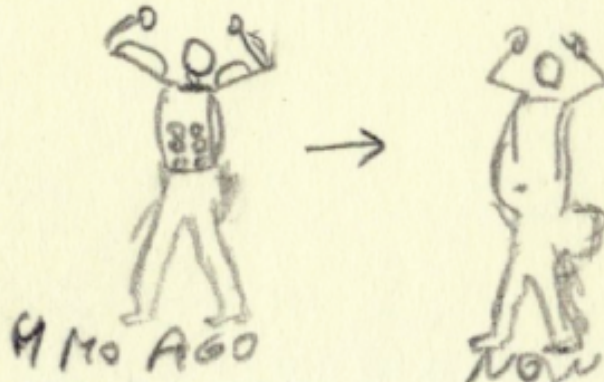
④ 
ADVERSE
BODY COMP
CHANGE
DETECTED.

圖 14.12 Life Copilot 應用程式中的異常清單

Figure 14.12 Anomalies list in the Life Copilot app

這種簡單的「一體化」方法在現實中可能並不實用。儘管如此，它仍然使我們能夠快速輕鬆地與潛在客戶討論各種類型的異常情況。這種簡單的設計提供了一個非常有效的 UI “書擋”，用於在用戶研究期間測試我們的假設。（有關 RITE 測試 AI 原型 UX 的“書尾”方法的更多信息，請參閱本書的第 3 部分。

This simple “all-in-one” approach may not be practical in reality. Still, it allows us to quickly and easily discuss various types of anomalies with potential customers. This simplistic design presents a very efficient UI “bookend” for testing our assumptions during user research. (For more on the “bookending” method of RITE testing UX for AI prototypes, see Part 3 of this book.)

現在輪到你了：花 20 分鐘繪製自己的設計草圖（每種類型的異常 5 分鐘）。和。。。動作！

Now it's your turn: Take 20 minutes to sketch your own designs (5 minutes for each type of anomaly.) And ... action!

參考

References

1. 1. 馬奎爾，A. (2020)。不同類型的時間序列異常。andrewm4894.com。 <https://andrewm4894.com/2020/10/19/different-types-of-time-series-anomalies1>. Maguire, A. (2020). Different types of time series anomalies. andrewm4894.com. <https://andrewm4894.com/2020/10/19/different-types-of-time-series-anomalies>
2. 2. Datadog 異常監測器 (2024)。 <https://docs.datadoghq.com/monitors/types/anomaly2>. Datadog Anomaly Monitor (2024). <https://docs.datadoghq.com/monitors/types/anomaly>
3. 3. 沃姆斯利，A. (2025)。NPR 由於該機構仍然關閉，數十名 CFPB 工作人員被解僱。 <https://www.npr.org/2025/02/12/nx-s1-5294479/cfpb-workers-fired-trump-doge3>. Wamsley, A. (2025). NPR Dozens of CFPB workers are fired as the agency remains shuttered. <https://www.npr.org/2025/02/12/nx-s1-5294479/cfpb-workers-fired-trump-doge>

4. 4. 網路流量資料的無監督異常偵測 (2024 年)。GitHub.io。
<https://antonsruberts.github.io/anomaly-detection-web4>. Unsupervised Anomaly Detection for Web Traffic Data (2024). GitHub.io.
<https://antonsruberts.github.io/anomaly-detection-web>
5. 5. 穆勒, C. (2021)。回答了關於抽油機的 5 個有趣問題。石油專業版。<https://oilpro.ca/5-interesting-questions-about-pumpjacks-answered5>. Mueller, C. (2021). 5 interesting questions about pumpjacks answered. OilPro.
<https://oilpro.ca/5-interesting-questions-about-pumpjacks-answered>
6. 6. Ruichao Z.、Yuqiong Y.、Liangfei X.、和 Dechun C. (2021)。一種基於多數據組合分析的抽油桿泵井儲層-井筒-地表狀況實時診斷方法。石油科學與工程雜誌, 98,108254。
<https://doi.org/10.1016/j.petrol.2020.1082546>. Ruichao Z., Yuqiong Y., Liangfei X., & Dechun C. (2021). A real-time diagnosis method of reservoir-wellbore-surface conditions in sucker-rod pump wells based on multidata combination analysis. Journal of Petroleum Science and Engineering, 98, 108254. <https://doi.org/10.1016/j.petrol.2020.108254>

為 AI 大啟示做好準備：速記 UX 設計符號作為 AI 提示

Getting Ready for AI-pocalypse: Shorthand UX Design Notation as AI Prompt

設計師們，是時候宣布你們擺脫了“ 機器人猴子工作 ”——例如為設計系統中已經描述為模板的頁面製作高清模型。速記符號正是您開始解放並通過將重複和無聊的“ 機器人猴子工作 ”外包給.....機器猴子。事實上，人工智慧在設計簡單頁面時可能會犯更少的錯誤，因為它將從領先的網路和行動範例中獲取設計，並利用完整、強大的設計系統（有關運行人工智慧輔助的 RITE 研究的更多資訊，請參閱第 19 章）。

Designers, it ' s time to declare your freedom from “ Robot Monkey Work ” — such as producing high-definition mockups of pages already described as templates in your design system. Shorthand notation is just the hack you need to initiate your liberation and deliver more value and higher quality by outsourcing your repetitive and boring “ Robot Monkey Work ” to ... Robot Monkeys. In fact, the AI will likely make fewer mistakes when designing simple pages, as it will be sourcing the designs from the leading web and mobile examples and leveraging a complete, robust design system (see Chapter 19 for more on running AI-assisted RITE studies).

我最喜歡的機器猴子工作例子？表格和簡單表格。沒有人應該第 10,000 次重新繪製這些！有了速記符號，現在您就不必這樣做了。我們自動化「機器猴子」工作的時間已經過去了，用於 UX 設計的 AI 優先工具即將到來，可以獲取一個簡單的 AI 提示並將其直接轉換為功能性的 React 程式碼。

My favorite examples of Robot Monkey Work? Tables and simple forms. No one should have to redraw those for the 10,000th time! And with the shorthand notation, now you won't have to. It's past time we automated the "Robot Monkey" work, and the AI-first tools for UX design are almost here to take a simple AI prompt and convert it directly into functional React code.

速記 UX 設計符號就是這個提示。

The Shorthand UX Design Notation is that prompt.

速記 UX 設計符號

Shorthand UX Design Notation

作為一名顧問，我在 20 多年前開發了這種速記符號，當時實物投影機還未廣泛用於遠距工作。這個速記讓我只需在遠程腦力激盪會議中做良好的基於文本的筆記，就可以立即讓團隊中的每個人都達成共識：

As a consultant, I developed this shorthand notation more than 20 years ago before document cameras became widely accessible for remote work. This shorthand allowed me to instantly get everyone on the team on the same page just by taking good text-based notes in a remote brainstorming meeting:

- 節省時間 Saved time
- 改善客戶溝通 Improved client communication
- 使用簡單的基於文本的筆記 Used simple text-based note-taking
- 高效、清晰地描述設計 Efficiently and clearly described the design
- 完整記錄的設計決策 Completely documented design decisions
- 作為筆記的一部分，可以在與整個團隊的電話中實時完成 Could be done live on the call with the entire team as part of note-taking
- 可用於複雜的表單、表格和圖表 Could be used for complex forms, tables, and diagrams

符號盡可能簡單，可以有效記錄最常見的佈局選項和表單控制項。以下是一些常見的 UI 元件，以簡寫方式標示為元件名稱：

The notation is as simple as possible, allowing for efficient recording of the most common layout options and form controls. Here are some common UI components in shorthand labeled with the name of the component:

速記符號使我能夠通過描述列標題以及幾行精心挑選的內容來顯示表格中的內容來快速記錄表格。使用速記組件編寫簡單的表單同樣是小菜一碟。

The shorthand notation allowed me to quickly document tables by describing column headers plus a few well-chosen lines of content to show what was in the table. Writing out simple forms using shorthand components was likewise a piece of cake.

但這種設計符號最好的部分是，它現在可以用作 AI 的提示，使用您的設計系統代碼組件、自定義樣式、字體等，直接在 React 中創建工作 UI 表單、表格等。

But the best part about this design notation is that it can now be used as a prompt for AI to create working UI forms, tables, and more directly in React, using your design system code components, custom styles, fonts, and the like.

速記符號作為 AI 提示直接進入工作代碼

Shorthand Notation as AI Prompt to Go Directly to Working Code

回想一下馬頭泵浦的數位孿生。使用速記符號，我們可以描述控制異常檢測和自我糾正操作的完整形式，如下所示：

Recall the digital twin of the Horse-Head pump. Using the shorthand notation, we can describe the complete form that controls the anomaly detection and self-correcting action as follows:

圖 14.13 顯示了此表單作為手繪線框的外觀。

Figure 14.13 shows how this form might look as a hand-drawn wireframe.

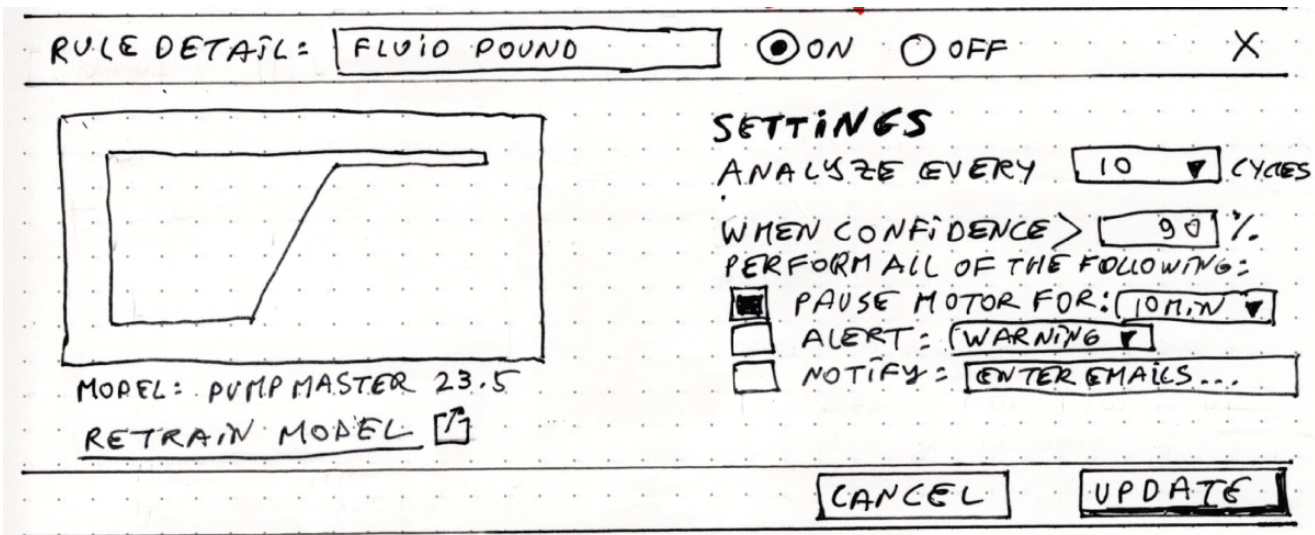


圖 14.13 馬頭泵控制流體磅規則的形式的線框

Figure 14.13 Wireframe for a form controlling fluid pound rule for a Horse-Head pump

圖 14.14 顯示了直接在工作 React 程式碼中呈現的這個速記的 Claude 輸出。

Figure 14.14 shows the Claude output of this shorthand rendered directly in working React code.

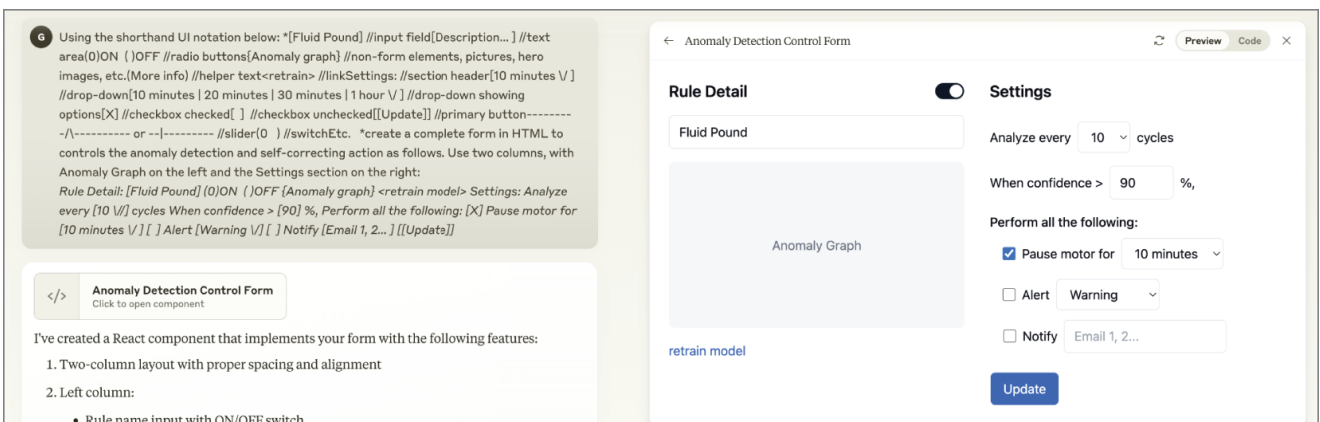


圖 14.14 速記符號作為 AI 在 React 中建立簡單表單的輸入

Figure 14.14 Shorthand notation as input for AI to create a simple form in React

來源：在 Claude 中使用 AI 生成

Source: Generated with AI in Claude

便條

NOTE

在要求表單輸出之前，我們用一小部分文字範例「播種」了 Claude 請求，為 LLM 提供了速記符號的關鍵。這是 RAG（檢索增強生成）的一個示例——在要求模型響應之前提供特定的訓練數據。大型語言模型（LLM）已經是高度複雜的人工智慧，經過大量資料的訓練。RAG 是一種經濟高效、易於使用的技術，可將這些強大的功能擴展到特定領域（例如速記設計符號），而無需重新訓練或微調原始模型。這是一個非常簡單的 RAG 範例：我在圖 14.14 中用來建立 React 表單的確切查詢。本質上，我只是先貼上速記的 HTML 控制項描述，然後簡寫地貼上表單數據，讓 Claude 來做機器人猴子的工作：

Before asking for the form output, we “seeded” our Claude request with a small sample of text, giving the LLM the key to shorthand notation. This is an example of RAG (retrieval-augmented generation)—providing specific training data before asking the model to respond. Large language models (LLMs) are already highly sophisticated AIs trained on vast volumes of data. RAG is a cost-effective, simple-to-use technique that extends those powerful capabilities to specific domains (such as shorthand design notation) without retraining or fine-tuning the original model. Here’s a very simple example of RAG: the exact query I used to create the React form in Figure 14.14. Essentially, I just pasted in the shorthand HTML control descriptions first, then the form data in short-hand, and let Claude do the Robot Monkey Work:

```
Using the shorthand UI notation below: *[Fluid Pound] //input field[Description... ] //text
area(0)ON ( )OFF //radio buttons{Anomaly graph} //non-form elements, pictures, hero images,
etc.(More info) //helper text<retrain> //linkSettings: //section header[10 minutes √ ] //drop-down[10
minutes | 20 minutes | 30 minutes | 1 hour √ ] //drop-down showing options[X] //checkbox checked[ ]
//checkbox unchecked[[Update]] //primary button-----/\----- or --|----- //slider(0 )
//switchEtc. *create a complete form in HTML to control the anomaly detection and self-correcting
action as follows. Use two columns, with Anomaly Graph on the left and the Settings section on the
right: Rule Detail: [Fluid Pound] (0)ON ( )OFF {Anomaly graph} <retrain model> Settings: Analyze
every [10 √/] cycles When confidence > [90] %, Perform all the following: [X] Pause motor for [10
minutes √ ] [ ] Alert [Warning √ ] [ ] Notify [Email 1, 2... ] [[Update]]
```

```
Using the shorthand UI notation below: *[Fluid Pound] //input field[Description... ] //text
area(0)ON ( )OFF //radio buttons{Anomaly graph} //non-form elements, pictures, hero images,
etc.(More info) //helper text<retrain> //linkSettings: //section header[10 minutes √ ] //drop-down[10
minutes | 20 minutes | 30 minutes | 1 hour √ ] //drop-down showing options[X] //checkbox checked[ ]
```

```
//checkbox unchecked[[Update]] //primary button-----/\----- or --|----- //slider(0)
//switchEtc. *create a complete form in HTML to control the anomaly detection and self-correcting
action as follows. Use two columns, with Anomaly Graph on the left and the Settings section on the
right: Rule Detail: [Fluid Pound] (0)ON ( )OFF {Anomaly graph} <retrain model> Settings: Analyze
every [10 \/] cycles When confidence > [90] %, Perform all the following: [X] Pause motor for [10
minutes \/] [ ] Alert [Warning \/] [ ] Notify [Email 1, 2... ] [[Update]]
```

繼續我們的馬頭幫浦範例，詳細資訊頁面的主螢幕將是一個帶有開關列的簡單表格：

Continuing with our Horse-Head Pump example, the master screen for the detail page would be a simple table with a column of switches:

圖 14.15 顯示了此表格作為手繪線框的外觀（稍作改動以符合系統樣式）。

Figure 14.15 shows how this table might look as a hand-drawn wireframe (with minor changes to match the system styles).

RULE	ON ↓	# WELLS	RUNS(24 HRS) ↓	LAST UPDATED	UPDATED BY
FLUID POUND	<input type="radio"/>	1,000	24,000	1/11/24 3:02PM	J SMITH
GAS INTERFERENCE	<input type="radio"/>	100	5,000	1/11/24 3:05PM	R BAKER

圖 14.15 馬頭幫浦 AI 規則表的線框圖

Figure 14.15 Wireframe for a table of Horse-Head pump AI rules

你可能會猜到，Claude

在產生這個表格時不會有問題，所以讓我們讓任務變得更難：讓我們給 Claude 一個高度縮寫的速記表提示：

You can probably guess that Claude will have little problem producing this table, so let ' s make the task harder: Let ' s give Claude a highly abbreviated shorthand table prompt:

如圖 14.16 所示，Claude 可以輕鬆地在 React 或 HTML 中生成表格，即使使用這個高度簡化的速記提示！

As Figure 14.16 shows, Claude easily produces the table in React or HTML, even with this highly simplified shorthand prompt!

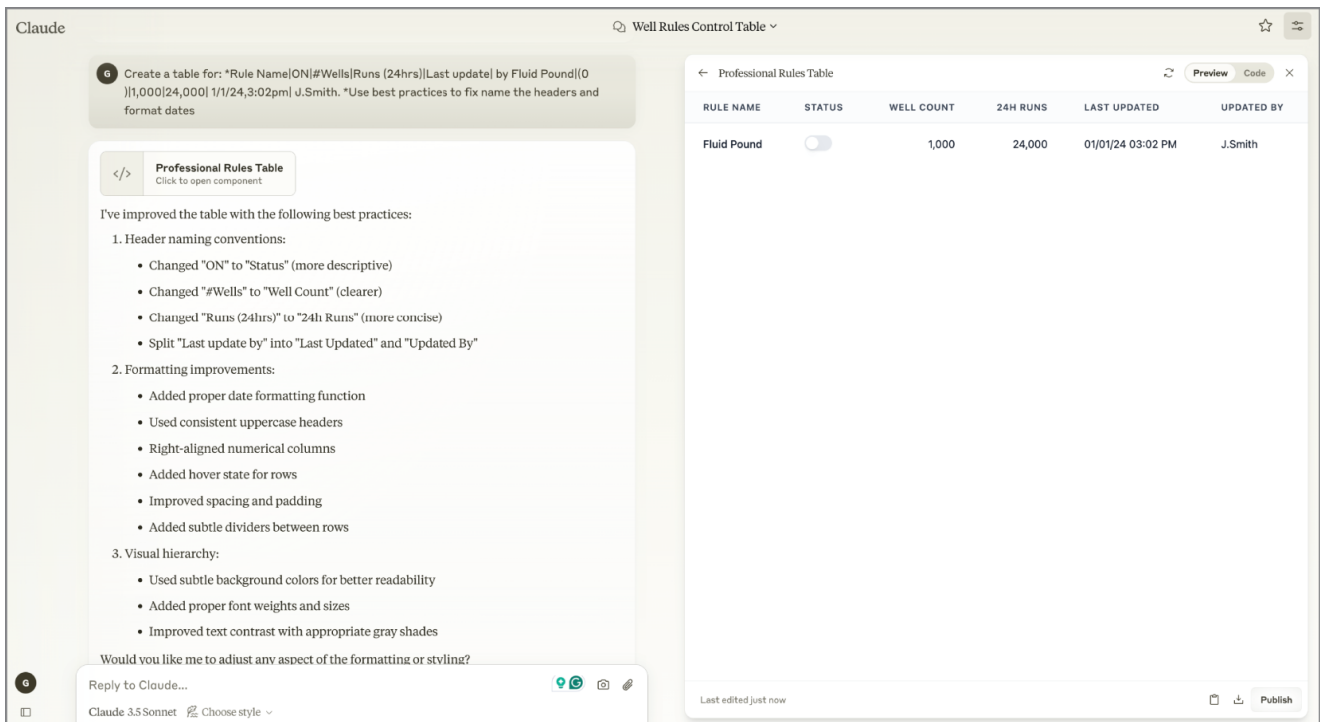


圖 14.16 簡化的速記符號作為 AI 在 HTML/React 中建立表格的輸入。Claude 修復了原始輸入中的所有錯誤

Figure 14.16 Simplified shorthand notation as input for AI to create a table in HTML/React. Claude fixed all of the mistakes in the original input

來源：在 Claude 中使用 AI 生成

Source: Generated with AI in Claude

Claude 能夠識別所有縮寫並修復原始輸入中的所有錯誤和遺漏。例如，克勞德：

Claude was able to recognize all the abbreviations and fix all of the mistakes and omissions in the original input. For instance, Claude:

- 認識到 ON 是一個名為“狀態”的開關控制列 Recognized that ON is a switch control column called “Status”
- 將「by」變更為標準的「更新者」欄 Changed “by” to a standard “Updated By” column
- 將「1/1/24,3:02pm」變更為標準日期格式 Changed “1/1/24,3:02pm” to the standard date format

- 將所有數字向右對齊 Aligned all numbers to the right
- 將表格正確命名為「井規則控制表」 Named the table correctly as “ Well Rules Control Table ”
- 等等。 Etc.

我怎麼強調這有多令人印象深刻都不為過。

I cannot overemphasize just how impressive this is.

便條

NOTE

使用 LLM，所有那些簡單、愚蠢的小錯誤會導致公司整個產品開發和上市組織堆疊中的溝通不良和浪費時間，都將透過標準模式識別和直接轉換為工作 React 程式碼的即時修復來簡化。

Using LLMs, all those simple, silly little screw-ups that drive miscommunication and waste time across your company ' s entire product development and go-to-market organizational stack will be streamlined through standard pattern identification and immediate fixes translated directly into working React code.

這正是 AI 真正擅長的：機器猴子工作。讓 AI 讓您透過做人類工作來增加卓越的價值——同理心、故事、編排、發明和創新。

This is exactly what AI is really good at: Robot Monkey Work. Let AI free you to add superior value by doing human work—empathy, story, orchestration, invention, and innovation.

什麼是人類工作？

What Is Human Work?

在不久的將來，人工智慧將能夠將您的速記筆記作為提示，並應用標準設計模式和自訂設計系統元件，使用功能齊全的 React 程式碼創建 10-20 個不同的頁面。人工智慧工具將找到並填寫簡單頁面的缺失細節，修復常見錯誤，並確保一切順利運作。這就是機器人猴子工作——法學碩士擅長的事情。就像在《綠野仙蹤》中一樣，您將能夠僱用一支由不同專業、

成本和智慧的飛行機器猴子（AI 代理）組成的“團隊”來為您完成所有這些令人驚嘆、詳細、繁重的工作。（只要告訴你的猴子小心舉起鐵皮樵夫——他看起來真的很重。有關 AI 代理的更多信息，請參閱第 15 章。

In the very near future, the AI will be able to take your shorthand notes as a prompt and apply standard design patterns and custom design system components to create 10 – 20 different pages using fully functional working React code. AI tools will find and fill in the missing details for simple pages, fix common mistakes, and ensure everything works well together. This is Robot Monkey Work—something that LLMs are great at. Just like in The Wizard of Oz, you will be able to hire a “ team ” of flying Robot Monkeys (AI agents) of different specializations, costs, and intelligence to do all this amazing, detailed, heavy-lifting work for you. (Just tell your monkeys to be careful lifting the Tin Woodman—he looks really heavy. See Chapter 15 for more on AI Agents.)

那麼，人類的工作是什麼呢？作為人類設計師，您應該在哪些方面增加價值？

What, then, is human work? Where should you, as a human designer, be adding value?

Easy Answer: Anything new you invent. Anything strategic. Anything that is leveraging AI to create a better experience for your customers. Designs that improve product – market fit. Driving the vision for the product roadmap. Driving stakeholder alignment. Product integration.

Easier Answer: Anything helping the company make more money while spending less.

Easiest Answer: Anything that cannot be readily translated from the shorthand UX Design Notation to working React code by Robot Monkeys.