Politecnico di Torino

Politecnico di Torino

MSc in Mathematical Engineering

A.y. 2024/2025

Graduation Session: October 2025

Vendor Performance Tracking and Cancellation Prediction in Amazon's Direct Fulfillment Operations

Integrating Dashboard Visualization and Time Series Models for Proactive Vendor Management

Supervisor: Eliana Pastor

Candidate: Martina Galfrè

Abstract

Efficient monitoring and forecasting of vendor performance are critical to the scalability and reliability of Amazon's Direct Fulfillment (DF) model, where third-party vendors ship products directly to customers without involving Amazon's own warehouses. This model requires reliable integration systems and timely order fulfillment across a broad network of vendors spread out over different regions.

This thesis aims to develop scalable methods to monitor vendor performance and integration health, and forecast vendor cancellations across Amazon's DF supply network. To achieve this, we designed and implemented the Vendor Health Dashboard, a centralized tool that automates monitoring of thousands of vendors worldwide, significantly reducing manual oversight and extending coverage to long-tail vendors who were previously difficult to track. For forecasting, we conducted a comparative evaluation of classical time series models, including ARIMA and ARMA-GARCH, alongside neural network models based on Gated Recurrent Units (GRUs). We assessed these models on their ability to predict cancellations, as well as their scalability and generalizability across different vendors and regions.

Results show that ARMA-GARCH models perform well for short-term forecasts involving vendors with high variance, while GRUs deliver higher accuracy and better generalization for longer-term predictions across various vendor types and regions. Notably, both models maintained strong performance when applied to unseen vendors without requiring custom modelling, supporting their scalability for large operational deployments.

These findings support a hybrid forecasting strategy that adapts to each vendor's behaviour, enabling teams to step in proactively and use data to better manage risks such as order cancellations. Bringing together monitoring and forecasting in a flexible dashboard gives a solid way to improve supply chain efficiency, minimize disruptions for customers, and support future improvements such as real-time alerts, additional metrics, and automated management of predictive models.

alla mia famiglia

Acknowledgements

I would like to express my heartfelt gratitude to the OSS Team at Amazon for making my internship such a rewarding and memorable experience. A special thank you goes to Andrei, who supported me through every step of the dashboard implementation. His positive attitude, deep understanding of the requirements, and help navigating EI processes made a real difference in bringing the dashboard to life. I'm also deeply thankful to Mikael, who played a crucial role during my first months on the team. He helped me get up to speed quickly, took the time to explain concepts with clarity and patience, and often encouraged me to take a step back and approach problems from new perspectives. Thanks to his guidance, working on the project was not only more insightful but also genuinely enjoyable. Lastly, I want to sincerely thank my manager, Matteo, the person who made this internship truly exceptional. His encouragement pushed me to grow, and his passion for his work and the people he leads has been truly inspiring. More than a manager, he has been a role model, someone I genuinely admire for his dedication, vision, and ability to bring out the best in those around him. His leadership style, grounded in passion and trust, created an environment where I felt both supported and motivated to push beyond my limits.

I would also like to thank my thesis advisor, Eliana Pastor, for her guidance throughout the process. Her feedback and thoughtful suggestions helped me shape a project that I am truly proud of.

Most important, I am deeply grateful to my family and friends. They have supported me in countless ways throughout this journey, celebrating the successes and helping me through the difficult moments. Their patience, encouragement, and belief in me gave me the strength to keep going, even when things felt overwhelming. This thesis would not have been possible without their love and support.

Table of Contents

List of Figures

Chapter 1

Introduction

In modern e-commerce, efficient inventory management and order fulfillment represent fundamental challenges, particularly for platforms handling millions of products across diverse categories. Traditional retail models require storing products in company-owned fulfillment centers before shipping to customers, an approach that can be costly and logistically complex for certain product categories. Direct Fulfillment (DF)[1] has emerged as an innovative solution to these challenges, representing a strategic partnership between e-commerce platforms and their vendors.

In the DF model, vendors maintain their own warehouses and fulfill customer orders directly, eliminating the need for intermediate storage. When a customer places an order, rather than shipping from a company warehouse, the order is routed to the appropriate vendor who ships the product directly to the customer. This process is seamless and transparent to the end user, who experiences the same level of service quality and reliability as with traditional fulfillment methods. Vendors may operate using their own shipping carriers or leverage the platform's carrier network, providing flexibility while maintaining control over the customer experience.

One of the companies which has employed the DF model is *Amazon.* Amazon is an American conglomerate which owns the worlds largest retail business in terms of market cap. Having started as an online book store Amazon has branched out into selling electronics, fashion articles, food and furniture amongst other things.

1.1 Focus and Scope

The success of the DF model depends not only on vendor reliability but also on seamless information exchange between Amazon and its vendors. This communication is managed by the *AmazonLink Integration, Operation and Intelligence*

(IOI) team, which oversees the electronic data interchange (EDI) and application programming interfaces (APIs) facilitating data flow between Amazon Retail and Direct Fulfillment vendors across North America, Europe, and Asia-Pacific. IOI is responsible for maintaining the quality and continuous improvement of these integration services, including documentation, notifications, training, and troubleshooting.

Despite these efforts, monitoring vendor performance across a global network of thousands of partners remains highly complex. Current monitoring is largely manual and resource-intensive, covering only a small fraction of vendors. For example, internal analysis showed that only 15–20 top EU vendors by volume receive detailed attention, requiring approximately 266 hours annually to complete 38 vendor reviews. This leaves thousands of vendors unmonitored, creating a critical visibility gap.

This thesis focuses on addressing these challenges by developing an automated Electronic Integration health monitoring dashboard that consolidates multiple data sources and tracks three fundamental performance metrics capturing key aspects of vendor health:

- Inventory Update (IAA) Errors: the ratio of errors to total items in vendor data feeds, measured over a rolling 60-day window, considering different error types;
- Order Fulfillment (OF) Delays: tracking processing delays between the assignment of an order to a vendor and its receipt;
- Order Fulfillment Cancellations: the ratio of order cancellations over the total number of shipments, with cancellations categorized into out-of-stock situations, full floor denials, item-level errors, and internal cancellations.

1.2 Relevance of Research

Monitoring vendor performance at scale is essential for ensuring operational efficiency and maintaining high customer satisfaction. Initial evaluations indicate that the current manual monitoring process covers only about 0.2% of vendors, leaving substantial operational risks unaddressed. Notably, internal reviews of the most important vendors in terms of business impact have revealed concerning patterns: 26% showed critical inventory accuracy issues and 21% faced significant cancellation problems. These findings suggest that unmonitored vendors may have even more severe challenges.

Poor vendor performance shows up in inventory updates inaccuracies, order fulfillment delays, and cancellations, all of which directly impact customer experience and supply chain operations. Although existing monitoring efforts are

labour intensive and inefficient, automation can dramatically improve scalability and responsiveness.

Moreover, despite the importance of forecasting vendor cancellations, current predictive capabilities are limited or non existent. Accurate forecasting methods would enable Amazon to prevent delayed or unfulfilled orders, which damage brand reputation and customer loyalty. They also generate inefficiencies in inventory management and supply chains, causing stock imbalances and logistical challenges. Without reliable forecasting, interventions happen only reactively, after cancellations occur, leading to costly operational disruptions.

Vendor behaviour varies widely across regions and product categories, complicating forecasting efforts and underscoring the need for robust, scalable analytics.

1.3 Research Questions and Objectives

This thesis aims to close these gaps by pursuing three main objectives. First, it aims to develop an automated vendor health monitoring dashboard that consolidates data from various sources to track key performance metrics continuously. Second, it seeks to design and rigorously evaluate forecasting models for vendor cancellations, utilizing both classical time series techniques and modern machine learning methods. Finally, the study explores how the insights gained through monitoring and forecasting can support vendor management teams in making informed decisions that improve operational efficiency and enhance customer satisfaction. In order to fulfill this purpose we propose the following research questions:

- How can vendor health monitoring be automated and scaled to cover thousands of partners globally?
- How do machine learning models, such as Gated Recurrent Units (GRUs), compare to classical time series models like ARIMA and GARCH in forecasting vendor order cancellations?
- Can predictive models accurately anticipate vendor cancellations sufficiently in advance to enable proactive interventions?
- How well do these models generalize across different geographic regions within Amazon's vendor network?

1.4 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 lays the theoretical foundations by outlining the mathematical and conceptual frameworks that support

the proposed approach. This chapter offers an overview of time series analysis techniques, including both classical econometric models and modern machine learning approaches. The classical econometric models include the Autoregressive Integrated Moving Average (ARIMA)[2] model, commonly used for time series forecasting, and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH)[3] model, which models time-varying variance. Modern approaches, such as Gated Recurrent Units (GRUs)[4], are also discussed for their ability to capture more complex temporal patterns.

Chapter 3 presents a literature review, examining the current state of the art and analysing existing applications of these models across diverse fields. Particular attention is paid to the adaptation of GARCH models, traditionally applied in financial market analysis, to the context of supply chain management. Similarly we summarize previous literature concerning the usage of GRUs in different fields.

The methodology, detailed in Chapter 4, describes the technical framework developed in this work. It begins with a description of the architecture and implementation of the vendor's health monitoring dashboard. This is followed by a discussion outlining how the time series models will be used. This includes data pre-processing steps, feature engineering, and considerations made in model design.

Chapter 5 presents the empirical results of the implemented solutions. This analysis covers both in-sample fit and out-of-sample predictive accuracy, with a particular focus on identifying prediction horizons that balance accuracy with operational needs. In particular, we look at horizons that would allow the IOI team to intervene timely in order to address potential vendor issues.

In Chapter 6, the discussion examines the experimental findings and their implications for managing vendor performance. It highlights how the developed predictive tools can be used to prevent order cancellations.

Finally, in Chapter 7, the thesis concludes by summarizing the main contributions. This closing chapter also outlines potential directions for future research, emphasizing opportunities to enhance and extend the current framework through further methodological and practical advances.

Chapter 2

Background

This chapter offers an overview of time series analysis techniques, covering both classical econometric models such as ARIMA and GARCH, as well as modern machine learning methods like Gated Recurrent Units.

2.1 Time Series Analysis

Autoregressive Moving Average (ARMA) models were introduced in 1976 by Box and Jenkins in their article [2]. Since then ARMA models have found a wide range of applications in modelling a stochastic process $(X_t)_{t\in\mathbb{Z}}$. A process is said to be an ARMA(s,t) process if it is a sum of an autoregressive part and a moving average part. That is if the process is of the form,

$$X_t = \sum_{i=1}^{p} \phi_t X_{t-i} + \sum_{i=1}^{q} \theta_i W_{t-i} + W_t,$$

with W_k being white noise and ϕ_k, θ_k being parameters. In order to model a process $(X_t)_{t\in\mathbb{Z}}$ as an ARMA we require the process to be weakly stationary. This, in part, means that the process to be modelled should have constant variance. For some processes this assumption is justifiable.

However, if the variance is varying with t, and this is due to the W_t term having changing behaviour, we can model this using Autoregressive conditional heteroskedasticity (GARCH) models, first introduced in [3]. In this case, we treat the heteroskedasticity not as an issue, but something to be modelled. We assume that the process to be modelled is of the form

$$\epsilon_t = \sigma_t w_t,$$

where σ_t is the time dependent standard deviation of the process and w_t is zero-mean white noise. More formally, a process $(\epsilon_t)_{t\in\mathbb{Z}}$ is said to be a GARCH(p,q) process if its two first moments exists finitely and satisfy, for $u < t$ with $t \in \mathbb{Z}$

1. $\mathbb{E}[\epsilon_t|\epsilon_u] = 0$,
2. $\exists \omega, \alpha_i, \beta_j$ with $i = 1, \ldots, q$ and $j = 1, \ldots, p$ s.t

$$\sigma_t^2 = \mathrm{Var}(\epsilon_t|\epsilon_u) = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2.$$

To model a process as an ARMA(s,t)-GARCH(p,q) we model the process conditional mean using an ARMA(s,t) model and the conditional variance using a GARCH(p,q) model.

2.2 Gated Recurrent Units

As described in [4], a Gated Recurrent Unit is a Recurrent Neural Network (RNN) [5] equipped with two additional gates to process information, an input gate and a forget gate. As opposed to Long Short-Term Memory (LSTM) Networks [6], GRUs do not have an output gate. This means that GRUs also have fewer parameters and hence are more computationally effective.

A GRU network consists of the input layer, implicit layers and finally the output layer. The hidden layers are composed of GRU neurons and the input is the data at time t. Suppose that the input sequence to the GRU-network is,

$$(x_1, \ldots, x_t)^T.$$

Then, at time t, the network conducts the following calculations, where $\odot$ denotes the Hadamard product:

$$r_t = \sigma(W_r \cdot \Big[h_{t-1}, x_t\Big]), \tag{2.1}$$
$$z_t = \sigma(W_z \cdot \Big[h_{t-1}, x_t\Big]), \tag{2.2}$$
$$n_t = \tanh(W \cdot \Big[r_t \odot h_{t-1}, x_t\Big]), \tag{2.3}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot n_t, \tag{2.4}$$
$$y_t = \sigma(W_o \cdot h_t). \tag{2.5}$$

Equations 2.1 and 2.2 represents the reset and update gate vector respectively. W_z is the weight matrix between the input and h_{t-1} in the update gate, where h_{t-1} is the output of the GRU unit at time $t-1$. W_r is the weight between the input and h_{t-1} in the reset gate. Equation 2.3 shows the calculation of the hidden state where W is the update gate's output. In the literature n_t is also called the candidate activation vector and essentially works as a substitute for the memory cell in an LSTM network. Equation 2.4 is how this hidden state is added to the current state,

z_t being the weight between the inputs. The last equation, Equation 2.5, is the final output of the GRU where W_o is the weight matrix for h_t, see [7].

The architecture of a GRU cell is shown in Figure 2.1, note that $\tilde{h}_t = n_t$. Intuitively, the update and reset gates manage the information flow on the GRU.

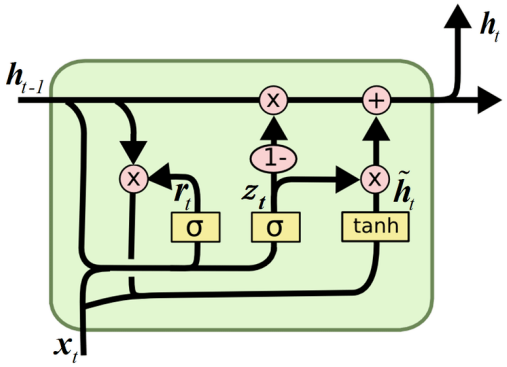


Figure 2.1: A GRU cell [8].

That is, it regulates what information the GRU should remember and which it should discard.

2.3 Bayesian hyperparameter optimization

Finding the most accurate hyperparameters for machine learning models is a research field itself. A grid search, where one looks through every combination of parameters and chooses the best model, is traditionally very expensive as a new model has to be trained and evaluated each time. A randomized grid search works in a similar way, but chooses only some of the parameters in the parameter space to look through at random. It is cheaper than a full grid search, as shown in [9], but at the cost of not looking through each parameter combination. Often the hyperparameters are chosen by some trial and error process. A cost-effective way to find appropriate hyperparameters, proposed in [10], is to use Bayesian optimization. In principle we want to solve the problem

$$\max_{x \in A} f(x)$$

and approximate the solution by building a surrogate for f. This surrogate is represented by $\mathbb{P}(\text{score} \mid \text{hyperparameters})$ and often modelled using Gaussian process regression. The surrogate is often much easier to optimize than the objective function, and we can use established Bayesian methods. These methods help in

finding the next set of parameters which maximizes the surrogate, these parameters are then evaluated on f and the surrogate is updated to incorporate the new results. Hyperparameters are chosen according to whichever gives the best score on the objective function. The broad idea of this means of hyperparameter search is essentially to make calculated guesses as to where the best hyperparameters are in the defined space, see also [11].

In this thesis Bayesian Optimization is used to find the optimal hyperparameters for the GRUs.

2.4 Metrics and Tests

This section outlines the key metrics and statistical tests used in this thesis to evaluate model performance and assess time series characteristics.

2.4.1 Root Mean Squared Error (RMSE)

The Root Mean Squared Error is a standard metric for evaluating the accuracy of predictive models [12]. It measures the square root of the average squared differences between predicted and actual values. Mathematically, for a set of predictions $\hat{y}_i$ and actual values y_i, where $i = 1, 2, \ldots, n$, RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{2.6}$$

RMSE gives more weight to higher errors due to the squaring operation, making it particularly sensitive to outliers. It is expressed in the same units as the dependent variable, which makes interpretation easier in the context of the data.

2.4.2 Mean Absolute Error (MAE)

The Mean Absolute Error is another common metric for assessing prediction accuracy [12]. Unlike RMSE, MAE measures the average of the absolute differences between predicted and actual values, giving equal weight to all errors. For predictions $\hat{y}_i$ and actual values y_i, MAE is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.7}$$

MAE is less sensitive to outliers compared to RMSE and is also expressed in the same units as the dependent variable. The comparison between RMSE and MAE can provide insights into the variance of errors; a larger difference between RMSE and MAE indicates a greater variance in individual errors.

2.4.3 Akaike Information Criterion (AIC)

The Akaike Information Criterion, introduced by H. Akaike in 1974 [13], is a measure used for model selection that balances model fit against complexity. It is particularly useful when comparing different models fitted to the same data. AIC is defined as:

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{2.8}$$

where k is the number of estimated parameters in the model and $\hat{L}$ is the maximum value of the likelihood function for the model. A lower AIC value indicates a better model, as it represents a better trade-off between goodness of fit and model simplicity.

For time series models like ARMA and GARCH, AIC helps in determining the optimal order (number of lags) by penalizing models with more parameters, thus mitigating the risk of overfitting. In the context of this thesis, AIC is used to compare different model specifications and select the most sparse model that captures adequately the characteristics of the data.

2.4.4 Augmented Dickey-Fuller (ADF) Test

The Augmented Dickey-Fuller test is a widely used statistical method for testing the presence of a unit root in a time series, which helps assess whether the series is stationary [14]. Stationarity, meaning the statistical properties of the series do not change over time, is a fundamental assumption for many time series models, including the ARMA family.

The ADF test builds upon the basic Dickey-Fuller test by including lagged differences of the series to account for higher-order serial correlation. The regression equation used in the ADF test is:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \delta_2 \Delta y_{t-2} + \ldots + \delta_p \Delta y_{t-p} + \epsilon_t \tag{2.9}$$

where $\Delta y_t = y_t - y_{t-1}$ is the first difference of the series, α is a constant (drift term), βt represents a deterministic time trend, γ is the coefficient of interest, and the δ_i terms account for lagged differences to mitigate autocorrelation. The error term ϵ_t is assumed to be white noise.

The null hypothesis of the ADF test is that the series has a unit root ($\gamma = 0$), implying non-stationarity. The alternative hypothesis is that the series is stationary ($\gamma < 0$). If the test statistic is more negative than the critical value, we reject the null hypothesis and infer that the series does not have a unit root.

In this thesis, the ADF test is employed to preliminarily assess whether the time series data is stationary before applying ARMA models, which require stationarity for reliable estimation and forecasting. It is important to note, however, that rejecting the null hypothesis does not guarantee stationarity. The ADF test

primarily checks for the presence of a unit root, which is a strong form of non-stationarity, but it may not detect other properties such as time-dependent variance. Conversely, if the null is not rejected, the series is almost surely non-stationary.

Therefore, while the ADF test provides valuable insight, it should be complemented with visual diagnostics such as time series plots, and autocorrelation (ACF) and partial autocorrelation (PACF) plots, to more robustly assess the stationarity of the series.

2.4.5 ARCH Test for Heteroskedasticity

The Autoregressive Conditional Heteroskedasticity (ARCH) test, also known as the Engle's ARCH test [15], is used to detect the presence of conditional heteroskedasticity (time-varying variance) in a time series. This test is particularly important when considering GARCH models, as these models are specifically designed to capture such patterns. Following the methodology described in the original article, the ARCH test is carried out in the following steps:

1. Estimate an autoregressive model for the time series and obtain the residuals $\hat{\epsilon}_t$.

2. Square the residuals and regress them on their own lagged values:

$$\hat{\epsilon}_t^2 = \alpha_0 + \alpha_1 \hat{\epsilon}_{t-1}^2 + \alpha_2 \hat{\epsilon}_{t-2}^2 + \ldots + \alpha_q \hat{\epsilon}_{t-q}^2 + \nu_t \tag{2.10}$$

   where q is the number of lags and ν_t is the error term.

3. Test the null hypothesis $H_0 : \alpha_1 = \alpha_2 = \ldots = \alpha_q = 0$ (no ARCH effects) against the alternative hypothesis that at least one $\alpha_i \neq 0$ (ARCH effects present).

The test statistic is nR^2, where n is the sample size and R^2 is the coefficient of determination from the regression in step 2. Under the null hypothesis, this statistic follows a chi-squared distribution with q degrees of freedom.

In this thesis, the ARCH test is employed to determine whether GARCH modelling is appropriate for the time series data. If the test rejects the null hypothesis, it indicates the presence of conditional heteroskedasticity, suggesting that GARCH models may provide better forecasts than models that assume constant variance.

Chapter 3

Related Work

In this chapter we present an extensive review of the current literature on ARIMA models, GARCH/ARCH models, and Gated Recurrent Units.

3.1 Previous Literature on ARIMA Models

ARIMA models have been central in time series analysis since their formalization by Box and Jenkins in [2]. These models have demonstrated their versatility across various domains due to their ability to capture linear temporal dependencies in stationary or differenced-to-stationary data.

In supply chain management, ARIMA models have been broadly applied for demand forecasting. In [16] Aburto and Weber combined ARIMA with neural networks to improve inventory management in a Chilean supermarket, demonstrating that hybrid approaches can outperform simple ARIMA models. Similarly, [17] proposed a hybrid ARIMA and support vector machines approach for stock price forecasting, showing improved accuracy over traditional methods.

For seasonal data, which is common in supply chains, Seasonal ARIMA (SARIMA) models have proven effective. Ediger and Akar in [18] applied SARIMA to forecast primary energy demand in Turkey, while [19] used ARIMA models to predict electricity prices in Spanish and Californian markets. Both studies highlighted ARIMA's capability to capture seasonal patterns and provide reliable short-term forecasts.

The comparative study by [20] evaluated various forecasting methods, including ARIMA, across different domains and found that ARIMA models consistently perform well for short-term forecasting tasks. However, they noted that performance degrades for longer forecast horizons, suggesting the need for more complex models in such scenarios.

Recent advancements have focused on addressing ARIMA's limitations. For

example, [21] demonstrated that hybrid approaches combining ARIMA with neural networks can significantly improve forecasting accuracy by capturing both linear and nonlinear patterns in time series data.

Despite the emergence of more sophisticated models, ARIMA remains relevant in modern forecasting applications due to its interpretability, established theoretical foundation, and computational efficiency. In [22] a comprehensive framework is provided for automatic ARIMA modelling, making these models accessible for large-scale forecasting applications.

3.2 Previous Literature on ARMA-GARCH

Much of the literature on GARCH is dominated by financial applications, see e.g. [23],[24],[25] and [26] for applications on stock market volatility, [27] for an application in electricity prices and [28] for an application on crude oil prices. This is mainly because these models were designed for such applications and take into account common patterns in financial data, like volatility clustering, varying levels of variability, and extreme values.

The work by Engle in [15] introduced the ARCH model to address time-varying volatility, which was later extended by Bollerslev [3] to the GARCH model. These models revolutionized volatility forecasting by explicitly modelling the conditional variance of time series data. The article [29] conducted a comprehensive comparison of GARCH-type models for volatility forecasting, finding that more complex variants like Exponential GARCH (EGARCH) and Glosten-Jagannathan-Runkle GARCH (GJR-GARCH) often outperform the standard GARCH model when asymmetric effects are present in the data.

Plenty of researchers have employed GARCH models outside of financial applications and seen good success. See for example [30] for an exotic application where the researchers attempt to predict the health of a machine using GARCH models. Similarly, GARCH models have been employed in the field of supply chains to predict e.g. inventory management and oscillated demand, see [31].

The authors of [32] use GARCH and EGARCH models in order to forecast production volatility with applications in supply chain management. The article finds that production volatility is time varying and often can be predicted.

In the context of supply chain disruptions, [33] applied GARCH models to quantify and forecast supply chain risks. In [34] this application is extended by combining ARMA-GARCH models with extreme value theory to predict extreme events in supply chains, which is particularly valuable for risk management.

The integration of ARMA and GARCH models (ARMA-GARCH) has proven particularly effective for time series that exhibit both autocorrelation in the mean and conditional heteroskedasticity. The authors of [35] applied ARMA-GARCH

models to electricity price forecasting, showing that accounting for both mean and volatility dynamics significantly improves forecast accuracy. Similarly, [36] demonstrated the effectiveness of ARMA-GARCH models for day-ahead electricity price forecasting in competitive markets.

Recent advancements include the development of multivariate GARCH models for capturing volatility spillovers across multiple time series. The paper [37] provided a comprehensive survey of multivariate GARCH models, highlighting their applications in portfolio optimization and risk management. In [38] practical aspects of implementing these models in high-dimensional settings are further explored.

The flexibility of GARCH models has led to multiple extensions tailored to specific data characteristics. For instance, [39] introduced the Fractionally Integrated GARCH (FIGARCH) model to capture long memory in volatility processes, while [40] proposed the EGARCH model to account for asymmetric volatility responses to positive and negative shocks.

3.3 Literature on Gated Recurrent Units

The GRU was first presented in [41] where the authors compared a Gated Recurrent Convolutional Network (grConv) to an RNN Encoder-Decoder for the purpose of translating text from English to French. The conclusion of this paper was that grConv's performance is comparable to that of the RNN Encoder-Decoder. Similarly, [42] compares different types of recurrent units in RNNs which employ some kind of gating mechanism. These models are evaluated and compared according to their performance in the tasks of modelling polyphonic music and speech signals. They also find that GRUs give results comparable to those of LSTM.

A comprehensive comparison between GRU and LSTM was conducted by [43], who evaluated over 10000 RNN architectures and found that GRUs can match or exceed LSTM performance while being computationally more efficient due to their simpler structure. This efficiency advantage was further explored by [44], who proposed a light GRU variant that reduces parameters while maintaining performance for speech recognition tasks.

In [45] the authors leverage GRUs in order to predict various financial sequences. As opposed to AMRA-GARCH models, the authors here comment that GRUs are rarely used in financial applications. The GRU is compared to a traditional deep net as well as a support vector machine model, of which the GRU outperforms both.

Building on this financial application, [46] demonstrated that GRU-based models have better results than traditional econometric models for stock market prediction by effectively capturing non-linear dependencies and long-term patterns. The

authors of [47] further showed that GRUs can effectively model the temporal dynamics of limit order books in high-frequency trading environments.

In [48] the authors attempt to predict resource demands on Cloud Workloads using GRUs. One of the arguments the authors give for choosing this model is its ability to perform well on time series data with long-range dependencies and high levels of noise. They find that, while GRUs by themselves perform well, their performance can be further improved using attention-based mechanisms.

The integration of attention mechanisms with GRUs has become increasingly popular. Li et al. proposed in [49] an independently recurrent neural network with attention for document classification, demonstrating significant improvements over standard GRUs. Similarly, [50] introduced a dual-stage attention-based GRU network for time series prediction, which outperformed traditional methods on multiple datasets.

The authors of [51] attempt to build a stacked model in order to forecast supply chain demand. They use, amongst other models, LSTM and GRUs. The findings indicate that GRUs, although outperformed by the combined model, outperform other models such as Stacked LSTM, vanilla LSTM, and Convolutional Neural Networks on the test task.

Similarly, [52] attempt to predict multivariate sales as a means of market forecasting for supply chain management. They build various models, including GRU, to predict the sales of a multitude of stores. The GRU is benchmarked against both Light Gradient Boosting Machine (LGBM) and Vectorized ARMA, and outperforms both. The authors attribute this to the GRUs ability to learn hidden patterns and its efficiency in handling temporal features.

In [53] the goal is to forecast product demand in supply chains. A model called GA-GRU is proposed, short for Genetic Algorithm (GA) with Gated Recurrent Unit. The reasoning is that GRU requires many parameters to be tuned for optimal performance, and the authors apply GA in order to optimize window size, number of neurons, initial learning rate, number of epochs and batch size. They compare their GA-GRU model to GRU, ARIMA and LSTM. The results indicate that, for their purpose, GRU is more powerful than LSTM. The authors also stress the importance of using robust hyperparameters that can be found using any number of different meta-heuristic methods.

The application of GRUs has extended beyond traditional time series forecasting. In the article [54] a GRU-based model for handling irregularly sampled time series in healthcare applications is proposed, demonstrating its effectiveness for clinical prediction tasks with missing data. The work of [55] further extended this approach with a non-autoregressive multiresolution GRU for imputing missing values in time series data.

Recent advancements include the development of hybrid models that combine GRUs with other techniques. Lin, Ye and Xu proposed in [56] a CNN-GRU model

for multivariate time series classification that uses CNNs for feature extraction and GRUs for temporal modelling. The authors of [57] introduced a graph convolutional GRU for traffic forecasting that incorporates spatial dependencies through graph structures, demonstrating superior performance over traditional time series models.

In the context of comparing GRUs with traditional statistical models like ARIMA and GARCH, [58] conducted a comprehensive evaluation for time series forecasting and found that GRUs consistently outperform statistical models for complex, non-linear time series. However, they noted that statistical models remain competitive for simpler, more linear time series, highlighting the importance of model selection based on data characteristics.

Chapter 4

Methodology

In this chapter we discuss the methodology of the thesis. It features two sections, one covering the methodology for the implementation of the DF EI health dashboard and one covering the forecasting part of the thesis.

4.1 Dashboard Implementation

The following subsections describe the key components of the dashboard. First, we explain how the three core performance metrics are computed, highlighting data sources, calculation logic, and aggregation methods. Next, we present the scoring system, which integrates these metrics into both overall and volume-adjusted scores to provide a comprehensive view of vendor performance. Finally, we provide an overview of the dashboard layout and its deep-dive sheets, showing how it supports both quick monitoring and in-depth analysis.

4.1.1 Metrics Computation

The vendor health monitoring dashboard focuses on three essential performance metrics that directly affect operational efficiency and customer satisfaction. Inventory Update (IAA) Errors, Order Fulfillment Delays, and Order Fulfillment Cancellations.

The first metric, IAA Errors, measures the percentage of errors in the vendor's inventory data feeds. In this context, a *feed* represents the inventory snapshot that a vendor submits to Amazon, detailing all products (identified by unique ASINs) and their quantities per warehouse. An ASIN, or Amazon Standard Identification Number, is a unique identifier assigned to each product listing. To calculate IAA errors, two key datasets are used. The first contains error records with timestamps and error types, reported at the ASIN level. The second provides counts of ASINs

processed correctly, also at the ASIN and warehouse level. Since there is no direct record of the total feed size, it is computed by summing the number of ASINs without errors (from the second table) and those with errors (from the first). This combined total reflects the feed size for a given warehouse and time period. The error percentage is then obtained by dividing the count of ASINs with errors by the total feed size. Aggregating this measure over a rolling 60-day window allows the dashboard to capture error trends and highlight prevalent issues.

For Order Fulfillment Delays, the objective is to track the time it takes for a DF vendor to receive an order after Amazon assigns it. Assignment timestamps, along with shipment IDs, are obtained from DF data. However, to measure the actual delay in the EDI transmission, that is critical for seamless vendor operations, it is necessary to link these shipments to their corresponding EI receipt dates. The EI receipt data includes transmission timestamps but lacks shipment identifiers. Therefore, a third table is utilized to connect EI transmissions with shipment IDs, enabling precise calculation of the delay between the order assignment and EI receipt. This measure reflects the vendor's responsiveness in processing orders within the electronic system, rather than the physical delivery time.

Finally, Order Fulfillment Cancellations are analysed by examining shipment records containing condition codes that indicate shipment status, including cancellations. These cancellations may arise from vendor-initiated actions or upstream issues such as inventory shortages or operational errors. By combining successful and cancelled shipment counts, the dashboard computes cancellation rates over a rolling 60-day window. This metric helps identify recurring bottlenecks and reliability concerns within the fulfillment process.

4.1.2 Scoring System

To effectively monitor vendor performance, the dashboard employs a scoring system based on three above mentioned metrics. These metrics are combined through weighted sums to produce two main scores: the Base Score and the Volume-Adjusted Score.

Let the metrics be denoted as:

$$m_1 = \text{IAA Error Rate}, \quad m_2 = \text{Cancellation Rate}, \quad m_3 = \text{OF Delay Rate}.$$

The OF Delay Rate, m_3, is calculated by considering only delays that exceed acceptable thresholds. Delays under 2 minutes are not included, as they are acceptable. Delays between 2 and 5 minutes are classified as yellow-flagged and weighted at 0.3, while delays exceeding 5 minutes are red-flagged and weighted at 0.7. These weighted components are combined to produce the overall OF Delay Rate used in the scoring system.

The corresponding weights, reflecting the relative importance of each metric, are set as:

$$w_1 = 0.5, \quad w_2 = 0.3, \quad w_3 = 0.2.$$

The *Base Score* S_{base} is calculated as the weighted sum of the raw metric values for each vendor:

$$S_{\text{base}} = w_1 m_1 + w_2 m_2 + w_3 m_3. \tag{4.1}$$

To put vendor performance in perspective within their regional context, each metric is normalized by the vendor's share of the total value of that metric across all vendors in the same region. Denote the normalized metrics as $\tilde{m}_i$ for $i = 1, 2, 3$, where each is computed as:

$$\tilde{m}_i = \frac{\text{Total Metric Value for Vendor}}{\sum_{\text{vendors in region}} \text{Total Metric Value}}.$$

For example:

- $\tilde{m}_1$ corresponds to the vendor's total number of inventory update errors divided by the regional total;
- $\tilde{m}_2$ is the number of cancelled shipments by the vendor divided by the total cancellations in the region;
- $\tilde{m}_3$ is the vendor's cumulative order fulfillment delay (in seconds) divided by the total delay observed across all vendors in the region.

The *Volume-Adjusted Score* S_{volume} incorporates these normalized metrics to account for both performance and scale:

$$S_{\text{volume}} = w_1 \tilde{m}_1 + w_2 \tilde{m}_2 + w_3 \tilde{m}_3. \tag{4.2}$$

These scores enable the user to identify vendors with critical performance issues both on an absolute scale and in relation to vendors from the same region, supporting targeted interventions.

4.1.3 Dashboard Overview and Deep-Dive Sheets

The vendor health monitoring dashboard comprises four main sheets, each designed to support different levels of analysis and operational decision-making:

- EI Health: this serves as the dashboard's command centre, providing an overview of vendor performance across all key metrics. It offers a high-level snapshot that facilitates rapid identification of vendors exhibiting performance issues or emerging trends requiring further investigation. In particular, it highlights the *top offender* vendors as identified by the scoring system, enabling quick prioritization of which issues to address first.

- IAA - Deep Dive: this sheet allows for a detailed examination of inventory update errors by categorizing error types such as invalid item ID, obsolete or suppressed item, and missing dropship flag. By breaking down these error patterns, it helps distinguish between systematic problems affecting multiple products or warehouses and isolated incidents.
- OF Delays - Deep Dive: this section provides a detailed temporal analysis of order fulfillment delays, presenting trends over time and distributions. It highlights the proportion of transmissions with quick responses (delays under 2 minutes, flagged green), moderate delays (2 to 5 minutes, flagged yellow), and significant delays (over 5 minutes, flagged red). This view helps to monitor vendor responsiveness.
- Cancellations - Deep Dive: this sheet offers an overview of cancellation behaviour by distinguishing between overall cancellation rates and vendor-specific cancellation rates, which only consider cancellations due to vendor errors. This distinction helps isolate vendor performance issues from upstream supply chain disruptions.

4.2 Time Series Analysis for Cancellation Forecasting

Building on the vendor health dashboard's ability to continuously rank vendors by performance across key metrics and regions, we identified the highest-impact vendors, the top offenders, in Europe, North America, and Far East. These vendors exhibit high cancellation rates and operational inefficiencies, making them good candidates for early intervention.

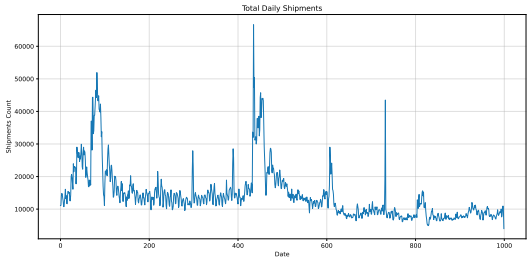
To better prevent and manage these risks, we developed a time series forecasting model focused on these key vendors. By concentrating on the top offenders in each region, the models can provide early warnings of possible cancellation spikes. This would help the IOI team to act quickly and prevent disruptions, and supports the dashboard's current scoring and monitoring tools.

4.2.1 Data

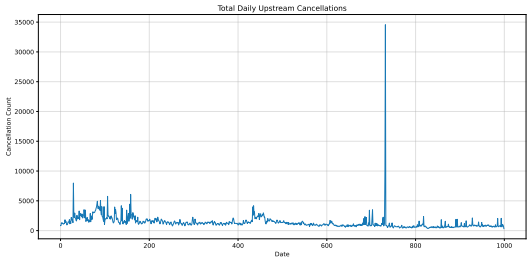
The dataset used for this analysis includes detailed daily shipment records from multiple warehouses, containing information such as order dates, shipment conditions, cancellation events, vendor and warehouse identifiers, timestamps, and status codes that indicate shipment progress or reasons for cancellation. From this raw data, we aggregated key metrics at the vendor level on a daily basis. Specifically, for each vendor and day, we compiled total shipment counts, vendor cancellation

counts, and upstream cancellation counts. These daily aggregated metrics form time series that capture shipment activity and cancellation patterns, which we then analysed to identify trends and potential correlations.

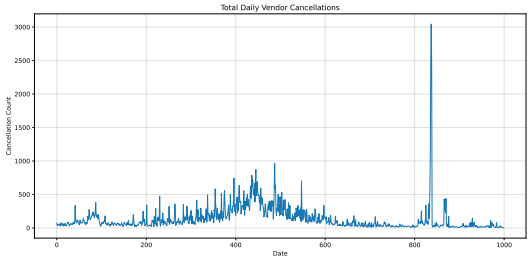
The plots in Figures 4.1, 4.2, and 4.3, illustrate the daily trends of the three key metrics, total shipments, upstream cancellations, and vendor cancellations, respectively for the North America, Europe and Far East top offenders. In several instances, we observe that spikes in shipment volume coincide with increases in cancellation counts, both on the vendor and upstream sides, suggesting a potential link between operational load and failure rates. These visual patterns motivate further analysis into how volume pressure may contribute to vendor performance issues. This observed relationship led us to explore a GRU model, rather than limiting the analysis to more traditional time series approaches such as ARIMA or ARMA-GARCH, which are univariate. As detailed in Subsection 4.2.4, the GRU model enables the use of multiple input features, including for example shipment volume, allowing for more flexible and data-driven forecasting of vendor cancellations.



(a) Total Daily Shipments (NA)



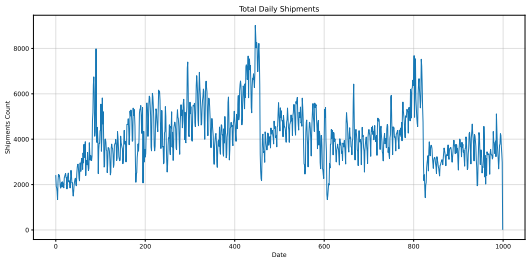
(b) Upstream Cancellations (NA)



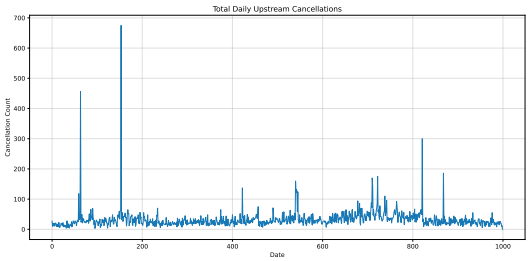
(c) Vendor Cancellations (NA)

Figure 4.1: Daily trends for the three key metrics: total shipments, upstream cancellations, and vendor cancellations for NA top offender.

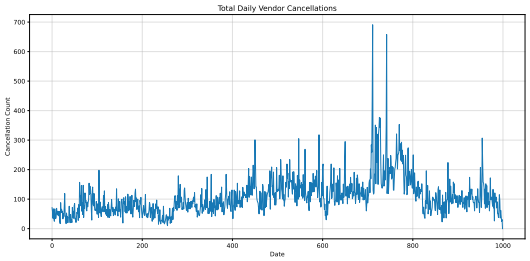
To further understand the temporal dependencies within the cancellation data, we examined the autocorrelation function and partial autocorrelation function plots of the original cancellation time series (see Figure 4.4). For clarity and interpretability, these plots refer specifically to the top offending vendor in the North



(a) Total Daily Shipments (EU)

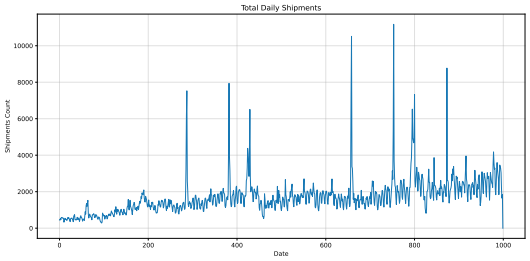


(b) Upstream Cancellations (EU)

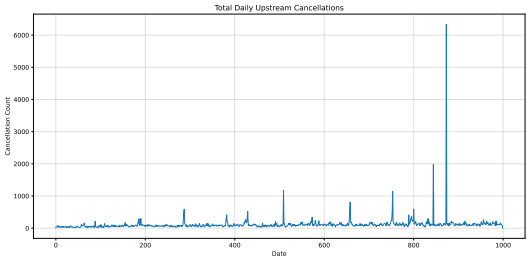


(c) Vendor Cancellations (EU)

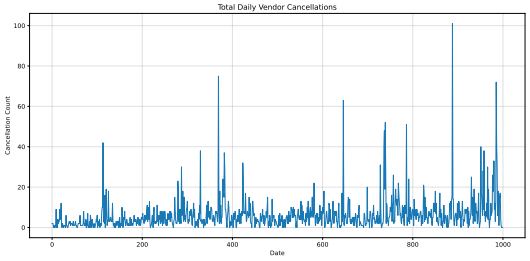
Figure 4.2: Daily trends for the three key metrics: total shipments, upstream cancellations, and vendor cancellations for EU top offender.



(a) Total Daily Shipments (FE)



(b) Upstream Cancellations (FE)



(c) Vendor Cancellations (FE)

Figure 4.3: Daily trends for the three key metrics: total shipments, upstream cancellations, and vendor cancellations for FE top offender.

America region. Visual inspection of these plots reveals a strong autocorrelation persisting for lags up to around 40 days, alongside a pronounced spike in the PACF at lag 1. This pattern suggests significant short- and medium-term dependencies in the data.

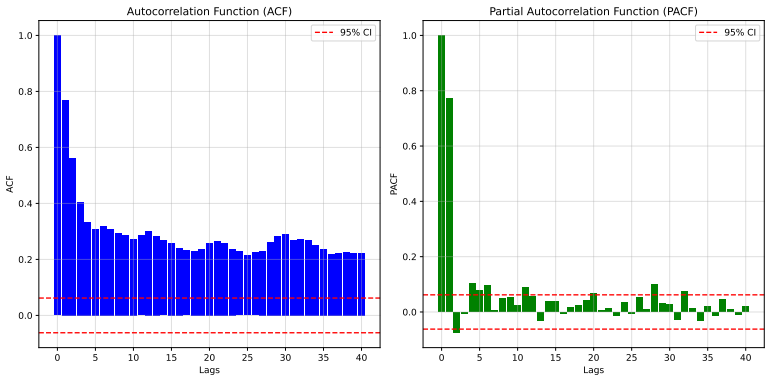


Figure 4.4: ACF and PACF of NA top offender.

Given the presence of strong autocorrelation and the evident non-stationarity, the cancellation series was transformed to stabilize variance and obtain stationarity. We applied a logarithmic transformation followed by first-order differencing, defined as:

$$x'_t = \log(x_t + 1) - \log(x_{t-1} + 1),$$

where adding 1 ensures the logarithm is defined even when the count is zero. The logarithmic transformation helps mitigate heteroskedasticity commonly observed in count data, while differencing removes linear trends. This transformation was applied only in the context of traditional statistical models, specifically ARIMA and ARMA-GARCH, which require stationary input series. In contrast, the GRU neural network model was trained on data transformed using the sklearn *MinMaxScaling*, as it does not rely on stationarity assumptions.

The transformed series was subsequently tested for stationarity using the Augmented Dickey-Fuller test, which yielded a highly significant p-value on the order of 10^{-14}, indicating strong evidence against the presence of a unit root. Furthermore, the ACF and PACF plots of the transformed series, shown in Figure 4.5, confirm that the preprocessing steps substantially reduced autocorrelation and stabilized the data. In particular, we see that both the ACF and PACF converge to zero following geometric decay.

To make the forecasts interpretable, the predicted values from models trained on transformed data were converted back to the original scale of cancellation counts by reversing the differencing and logarithmic transformations. This was performed in two steps: first, the differenced forecast $\hat{x}'_t$ was added to the previous

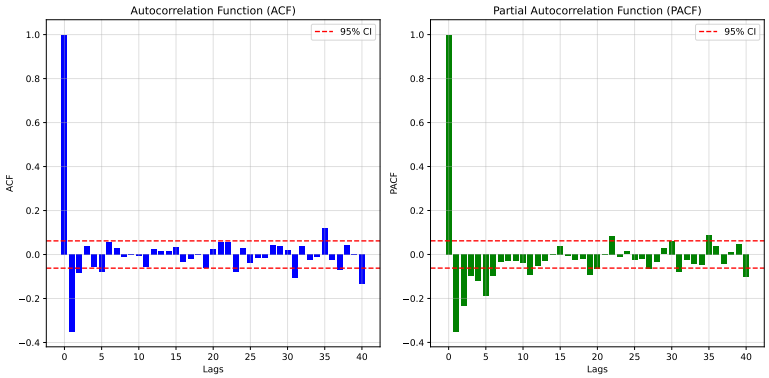


Figure 4.5: ACF and PACF of the transformed temporal series.

log-transformed observed value:

$$\hat{x}_t = \hat{x}'_t + \log(x_{t-1} + 1),$$

then, the original scale was recovered by exponentiation:

$$\hat{x}_t^{\text{orig}} = \exp(\hat{x}_t) - 1.$$

This procedure produced forecasted values expressed as actual daily cancellations, enabling their direct use in decision-making.

For all models, we split the data into training and testing sets, using 80% for training and the remaining 20% for testing. This division allows us to evaluate how well the models perform on data they have not seen before, which helps to estimate their accuracy in real-world forecasting. To keep the comparison fair, we calculated all evaluation metrics on the original data scale.

4.2.2 ARIMA Model

The ARIMA model was selected as a baseline to benchmark the performance of more advanced forecasting approaches. As a widely used and well-understood method for modelling univariate time series, ARIMA provides a reference point for evaluating the added value of more complex models.

To identify an appropriate ARIMA configuration, a grid search was performed over the following hyperparameter ranges:

$$p \in \{0, \ldots, 9\}, \quad d \in \{0, 1\}, \quad q \in \{0, \ldots, 9\}.$$

Candidate models were assessed based on multiple criteria including RMSE, MAE, and AIC. The final model was selected as the one with the lowest AIC.

After fitting the ARIMA model, we checked the residuals to evaluate how well the model captured the dynamics of the data. In particular, we looked at the squared residuals over time, shown in Figure 4.6, as sustained deviations from zero in these values can indicate heteroskedasticity. To test this more formally, we used the ARCH test, which confirmed that the residuals exhibited significant ARCH effects. Based on this result, we decided to extend the analysis using an ARMA-GARCH model, which is better suited to handle time series with time-varying variance. Further details are provided in the next subsection.

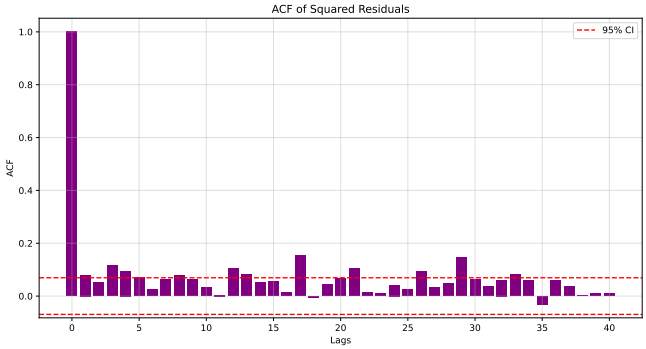


Figure 4.6: ACF of the squared residuals from the ARIMA model.

4.2.3 ARMA-GARCH Model

To capture both the autocorrelation structure and the time-dependent variance observed in the transformed cancellation time series, an ARMA-GARCH framework was adopted. This modelling approach allows to model the conditional mean using an ARMA process, while handling changing variability in the residuals with a GARCH component.

In particular, the conditional mean was modelled using an ARMA(s, t) process, paired with a GARCH(p, q) specification for the variance. The differencing order was kept at zero, as the input series had already been log-transformed and differenced during preprocessing.

The modelling proceeded in two stages: first, an ARMA(s, t) model was fitted to the transformed training data to capture the conditional mean,

$$y_t = \mu_t + \epsilon_t, \quad \mu_t = \text{ARMA}(s, t).$$

Then, the residuals from this fit were modelled using a GARCH(p, q) process, to

capture time-varying volatility:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2.$$

Forecasts over the entire test horizon were generated as:

$$\hat{y}_t = \hat{\mu}_t + \epsilon_t, \quad \epsilon_t \mid \mathcal{F}_{t-1} \sim \mathcal{N}\Big(0, \sigma_t^2\Big)$$

where $\hat{\mu}_t$ and $\hat{\sigma}_t^2$ denote the predicted conditional mean and variance at time t.

Similarly to the baseline model, a grid search was conducted over the ARMA orders (s, t) and GARCH orders (p, q) within the set $\{1, 2, 3, 4\}$. The best model was chosen on the basis of the lowest AIC.

To better mimic real-world forecasting, we applied a rolling forecast approach using the selected ARMA-GARCH model. Starting with the training data, the model was repeatedly updated as new observations became available. At each step, forecasts were made for a fixed horizon (e.g. 1, 3 or 30 steps ahead) by fitting the ARMA model to the current data to get the conditional mean, and then fitting the GARCH model to the residuals to estimate conditional variance. The forecasts were computed as

$$\hat{y}_{t:t+h} = \hat{\mu}_{t:t+h} + \epsilon_{t:t+h}, \quad \epsilon_{t:t+h} \sim \mathcal{N}\Big(0, \hat{\sigma}_{t:t+h}^2\Big).$$

where $\hat{\mu}_{t:t+h}$ is the predicted mean and $\hat{\sigma}_{t:t+h}^2$ is the estimated variance over the forecast horizon.

After each forecasting step, the actual observed values were added to the dataset before proceeding to the next iteration. This rolling setup lets the model adapt to new data over time, making the forecasts more realistic. Finally, the predicted values were transformed back to the original scale of cancellation counts for interpretation.

4.2.4 GRU-Based Deep Learning Model

The earlier plots revealed strong relationships between shipment volumes, upstream cancellations, and vendor cancellation counts, especially a pronounced connection between vendor cancellations and total shipments. This suggests that vendor cancellations are driven not only by their own historical patterns but also by multiple interacting factors that evolve over time, potentially showing nonlinear dependencies and long-range temporal effects. To further explore these relationships, we calculated Spearman's rank correlation coefficients, which measure monotonic associations between variables without assuming linearity (see more in [59]). This analysis confirmed the presence of significant monotonic trends among the features,

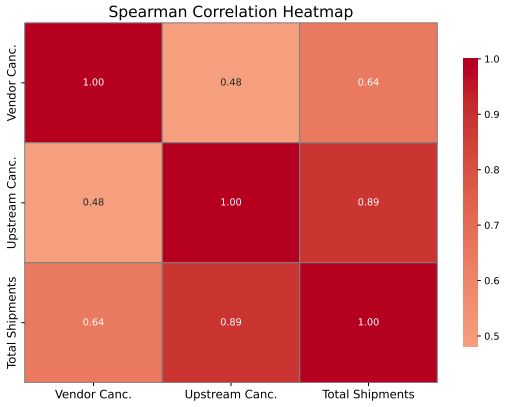


Figure 4.7: Spearman correlation heatmap between vendor cancellations, upstream cancellations, and total shipments.

supporting the need for models capable of capturing complex, nonlinear temporal dependencies.

As illustrated in Figure 4.7, these monotonic correlations further justify the adoption of advanced sequence modelling techniques. To effectively capture these complex dynamics, we adopted a deep learning approach based on Gated Recurrent Units.

Unlike traditional linear models, which often assume fixed relationships and limited memory of past values, GRUs provide the flexibility to model sequences with multiple correlated features and can capture nonlinear patterns more effectively. Our model used the following input features: vendor cancellation counts, upstream cancellation counts, and total shipment volumes, together with temporal features. Prior to training, all non-temporal features were normalized to the [0,1] range using Min-Max scaling, ensuring the model treats all inputs on a comparable scale, which stabilizes and accelerates learning.

To prepare the data for training, we employed a sliding window method that converts the multivariate time series into supervised learning examples. Each input consists of a sequence of 50 consecutive time steps of the features, and the corresponding target is the vendor cancellation count immediately following this sequence. This setup formulates the task as a many-to-one forecasting problem, where the model learns to predict the next vendor cancellation count based on historical feature data. Additionally, the day of the week corresponding to the target time step is included as a predictive feature, enabling the model to capture weekly patterns in cancellation behaviour.

The architecture consists of multiple stacked GRU layers, with the number of hidden units varied as part of hyperparameter tuning. After the recurrent layers, a dropout layer, with a fixed dropout rate $\rho = 0.2$, was included to reduce overfitting

by randomly deactivating neurons during training. Finally, a dense layer outputs a single scalar representing the forecasted vendor cancellations for the next time step. We limit the training to 100 epochs with early stopping. The callback for early stopping is validation loss together with a patience of 3. This means that 15% of the training sample is used for validation during training and if the validation loss increases 3 times the training of the parameters are halted.

Given the many hyperparameters affecting model performance, i.e. the number of GRU units, number of stacked layers, batch size, and number of epochs, careful tuning is essential. Instead of relying on traditional grid search, as previously mentioned we employed Bayesian hyperparameter optimization. This approach uses Gaussian processes to create a probabilistic model of performance, helping to efficiently find the best hyperparameters.

To better understand the optimization process and assess the influence of each hyperparameter, we visualized the search results using *Optuna*'s built-in tools. In particular, the slice plot, shown in Figure 4.8, depicts the distribution of objective values (i.e. mean squared error) across the entire range of each hyperparameter by plotting the observed performance at different sampled values, for the NA vendor. This allows us to examine how changes in individual parameters impact model performance while marginalizing over other parameters, revealing trends such as regions where increasing a parameter improves or worsens results. If performance clustered near the upper or lower bounds of a parameter (e.g., consistently better results with higher units), it indicated that the current range might need to be expanded or shifted. This iterative analysis helped guide adjustments to the parameter ranges leading to improved model performance.

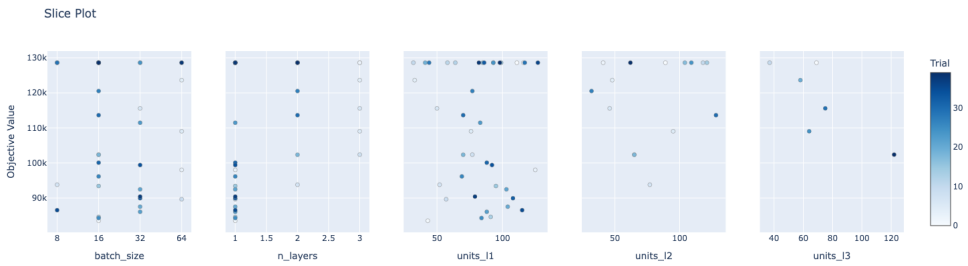


Figure 4.8: Slice plot showing how individual hyperparameters affect model MSE.

The full hyperparameter space searched through is the following:

$$\begin{aligned} \text{n\_layers} &\in \{1, 2, 3\}, \\ \text{units\_per\_layer} &\in \{32, \ldots, 128\}, \\ \text{batch\_size} &\in \{8, 16, 32, 64\}, \\ \text{epochs} &= 40, \\ \text{dropout\_rate} &= 0.2. \end{aligned}$$

The activation function for hidden layers is given by the hyperbolic tangent,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

and the activation function at the output layer is given by the Rectified Linear Unit (ReLU),

$$\text{ReLU}(x) = \max(0, x).$$

We use the Adaptive Moment Estimation (Adam) optimizer with default parameters [1], and MSE as the loss function. The Adam optimizer works by using parameter specific learning rates based on 2^{nd} order moments, see e.g. [60]. We proceeded to evaluate the model using multiple forecast horizons: 1, 3, and 30 days to enable direct comparison with the time series models. Forecast accuracy was evaluated using the aforementioned metrics: MSE, RMSE, and MAE.

4.2.5 Scalability of the Models

To evaluate the scalability of the proposed forecasting approach, we conducted a generalization experiment in which the best-performing models for each region were applied to additional vendors within the same geographic area. Specifically, we tested both ARMA-GARCH models and GRU models, using the hyperparameters previously optimized for the top-performing vendor in each region.

This setup reflects a realistic operational scenario: rather than tuning separate models for each vendor, we assess whether a model trained and optimized for one vendor can be reused for others in the same region without further modification. This form of intra-regional generalization is particularly important for scalable deployments, where resource constraints make individual model tuning impractical.

While this is not generalization in the strict machine learning sense of training and testing across independent and identically distributed samples, it offers an approximation of how models would perform in production across a heterogeneous

[1] https://keras.io/api/optimizers/adam/. *Accessed* 2025-07-20.

vendor base. It also reflects the organizational structure of vendor management at Amazon, where teams are typically responsible for specific regional portfolios.

The experiment was carried out using the same evaluation setup and timeframes described in previous sections, ensuring consistency in the comparison of GRU and ARMA-GARCH models.

Chapter 5

Results

This chapter presents the results of this thesis. It is structured into two main sections, reflecting the two core components of the work. The chapter starts with a few comments about the DF EI Health Dashboard. The second section presents the development and evaluation of the predictive models aimed at forecasting vendor cancellations.

5.1 Vendor Health Dashboard Results

The first element of the dashboard, shown in Figure 5.1, is a summary table offering an overview of vendor performance across the three core metrics: IAA Error Rate, Average OF Delay, and Cancellation Rate. Each vendor is scored and flagged using a traffic-light system: green for good performance, yellow for moderate issues, and red for critical cases, enabling quick identification of vendors requiring attention. For confidentiality, information such as company codes, vendor codes, and warehouse codes has been removed from this figure and from all subsequent dashboard views in this section.

Alongside the metrics, the table includes other descriptive attributes of vendors. These include company and vendor codes and names, warehouse identifiers, and the region and marketplace of operation, helping localize and segment performance. Financial and operational attributes such as General Ledger (GL) classification, Gross Merchandise Sales (GMS), and shipment volume provide insight into the vendor's scale and impact on Amazon business. The table also reflects the vendor's level of integration through the EI profile (e.g. EDI, API, hybrid, or not integrated) and the IAA feed type (automated via EDI/API or manual via *Vendor Central*, an Amazon internal portal). Labelling practices are captured as well, distinguishing between vendors using Amazon-provided labels and those shipping through their own carriers (Vendor Own Carrier, VOC).

EI Health Summary Table

Company Code	Company Name	Vendor Code	Vendor Name	GL	Region	Marketplace	EI Profile	EI IAA Feed	Label Type	GMS	Shipments Volume	IAA Error Rate	OF Delay	Cancellation Rate
				BISS	EU	FR	API	VC	Amazon	$7,929,280	92,499	62%	00:00:04	3%
				Lawn and...	EU	BG	EDI	EDI/API	Amazon	$5,933,177	88,462	75%	00:00:46	3%
				BISS	EU	DE	API	VC	Amazon	$5,356,137	66,949	28%	00:00:04	3%
				Large ...	EU	DE	EDI	EDI/API	Amazon	$5,146,502	8,005	18%	00:00:37	1%
				Large ...	EU	DE	EDI	EDI/API	Amazon	$5,144,453	7,979	8%	00:00:37	1%
				Books	EU	DE	EDI	EDI/API	Other	$4,447,201	195,680	6%	00:00:45	3%
				Furniture	EU	DE	API	VC	Amazon	$3,453,892	122,671	14%	00:00:04	2%
				Furniture	EU	ES	HYBRID	VC	Amazon	$3,232,779	34,985	2%	00:00:04	4%
				Furniture	EU	DE	EDI	EDI/API	VOC	$3,207,257	45,787	2%	00:00:43	2%
				Kitchen	EU	DE	EDI	EDI/API	Amazon	$3,182,827	23,610	19%	00:00:43	2%
				BISS	EU	BG	API	VC	Amazon	$2,901,431	47,457	32%	00:00:04	3%
				BISS	EU	IT	API	VC	Amazon	$2,833,043	34,775	51%	00:00:04	5%
				Furniture	EU	BG	HYBRID	VC	Amazon	$2,565,939	44,411	6%	00:00:04	1%
				Kitchen	EU	DE	EDI	EDI/API	Amazon	$2,498,443	13,518	20%	00:00:43	2%
				Home ...	EU	BG	NOT ...	VC	Amazon	$2,423,456	94,806	31%	00:00:04	2%
				Personal ...	EU	DE	EDI	EDI/API	VOC	$2,141,591	15,821	31%	00:00:53	4%
				Personal ...	EU	BG	EDI	EDI/API	Amazon	$1,854,079	8,138	74%	00:00:43	5%
				Furniture	EU	BG	API	VC	Amazon	$1,713,852	15,071	46%	00:00:04	4%
				BISS	EU	DE	API	VC	Amazon	$1,628,202	8,640	0%	00:00:04	3%
				Lawn and...	EU	FR	NOT ...	VC	Amazon	$1,570,775	30,066	12%	00:00:04	2%
				Lawn and...	EU	DE	EDI	VC	Other	$1,495,195	93,300	0%	00:00:41	2%
				Furniture	EU	DE	API	VC	Amazon	$1,409,535	14,726	44%	00:00:04	2%
				Home	EU	BG	EDI	EDI/API	Amazon	$1,390,112	16,550	6%	00:00:41	2%
				Furniture	EU	IT	HYBRID	VC	VOC	$1,380,283	10,559	0%	00:00:04	6%
				Music	EU	BG	API	VC	Amazon	$1,337,320	59,138	27%	00:00:04	1%
				Automotive	EU	DE	EDI	EDI/API	Amazon	$1,314,897	11,374	72%	00:00:39	22%

Figure 5.1: DF EI Health Summary Table, displaying attributes and metrics for each vendor-warehouse combination.

To quantify and prioritize vendor risk, the dashboard includes a scoring table (Figure 5.2) that brings together both performance metrics and business impact. Each vendor is assigned a total score, with individual components reflecting their behaviour across key EI dimensions. The table includes two scoring columns: one that evaluates performance in isolation, computed using the base scoring formula (Equation 4.1), and one that adjusts this score to account for the vendor's impact on the overall business, incorporating factors such as sales volume and shipment activity, as defined in the volume-adjusted scoring formula (Equation 4.2). This dual perspective enables stakeholders to identify vendors who may seem operationally reasonable but whose issues have a larger effect due to their business scale. As a result, the system helps direct attention toward those vendors whose behaviour poses the greatest risk to the supply chain.

One key component of vendor performance is the correctness and consistency of item data contained in the inventory feeds. Figure 5.3 highlights the most common errors found among flagged vendors, including invalid item IDs, obsolete or suppressed items, and missing dropship flags.

The dashboard also monitors Order Fulfillment delays, which can impact overall operational efficiency. Figure 5.4a shows how these delays have evolved over time for vendors flagged as high risk. Figure 5.4b further categorizes delays by severity: orders fulfilled in under 2 minutes, between 2 and 5 minutes, and over 5 minutes.

The final component of the dashboard focuses on cancellations, one of the most disruptive issues in vendor operations. Figure 5.5 distinguishes between cancellations initiated by vendors themselves and those occurring upstream (e.g. due to system issues). For several vendors flagged in red or yellow in the cancellation

Vendor Performance Impact Assessment

Cumulative risk score table (0-100), with and without considering orders volume
Higher scores indicate greater risk or poorer performance

Vendor Code	Region	Marketplace	Vendor Score (No Volume)	Vendor Score
	EU	DE	3.91	12.84
	EU	BG	21.7	11.52
	EU	DE	3.54	2.82
	EU	DE	42.45	2.23
	EU	IT	26.54	2.18
	EU	IT	38.32	2.04
	EU	ES	37.54	2.01
	EU	FR	37.63	1.96
	EU	DE	46.76	1.84
	EU	BG	38.41	1.83
	EU	BG	17.87	1.65
	EU	ES	37.94	1.56
	EU	EG	24	1.56
	EU	DE	0.69	1.53
	EU	DE	46.75	1.51
	EU	BG	0.75	1.4
	EU	DE	42.64	1.16

Figure 5.2: Vendor Scoring Table including Base-Scores and Volume-Adjusted Scores.

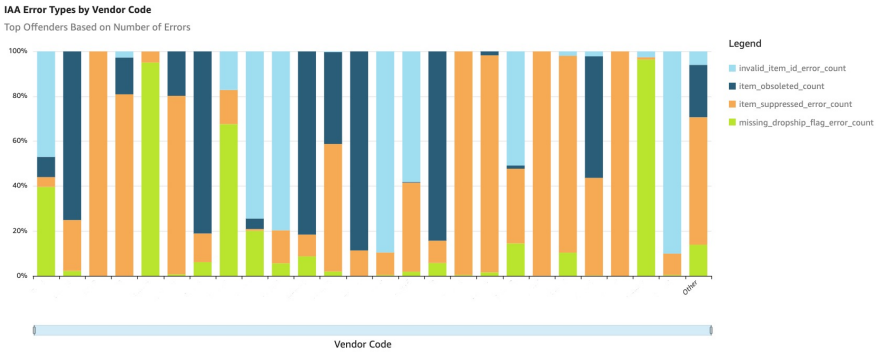
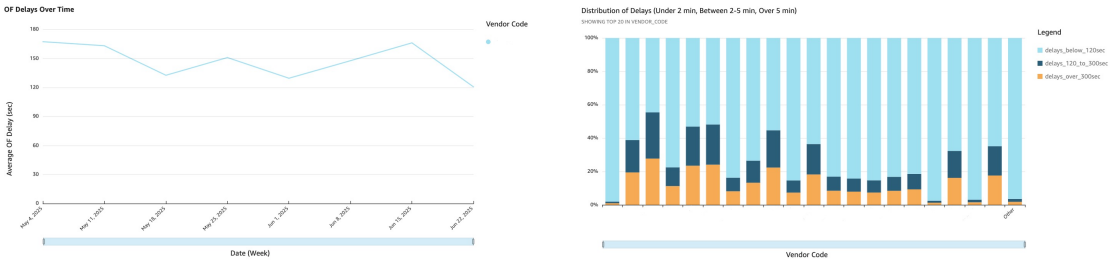


Figure 5.3: IAA error types across flagged vendors.

column of the summary table, vendor-initiated cancellations frequently exceed upstream ones. Since these cancellations directly harm vendor experience, predicting them ahead of time becomes important.

In the following section, we present the results of the predictive models focused on vendor cancellations. These models aim to forecast disruptive events before they happen. By identifying likely cancellation spikes in advance, we can support timely interventions and improve overall vendor performance.



(a) OF Delays Over Time **(b)** Distribution of OF Delays

Figure 5.4: Visualization of OF delays for flagged vendors.

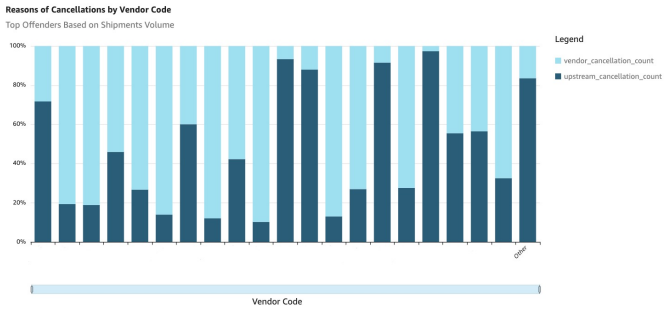
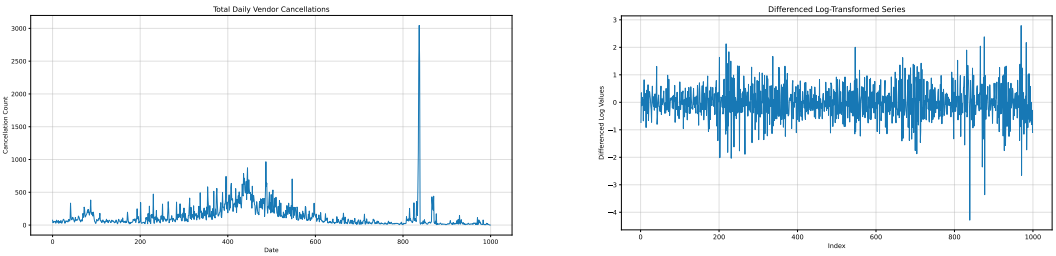


Figure 5.5: Comparison of Vendor and Upstream Cancellations.

5.2 Cancellation Forecasting Results

Following the procedure outlined in the methodology, we begin the modelling by transforming the original vendor cancellation count series to stabilize variance and prepare it for time series analysis. The left panel of Figure 5.6 displays the raw cancellation count over time, showing noticeable variance and potential non-stationarity. In the right panel, the transformed series is shown after first-order differencing, which helps remove trends and stabilize the mean. This differenced log-transformed series serves as the input for the ARIMA and ARMA-GARCH models.

The cancellation forecasting was implemented for the top offender in each of the three regions, North America, Europe, and the Far East. However, for clarity and to highlight key observations that motivated further model refinement, we present the NA vendor as a representative case. The same analytical steps and diagnostics were applied to the other two vendors, and their respective performance metrics and forecast plots are reported in the final comparison tables.



(a) Original vendor cancellation count **(b)** Differenced Log-transformed series

Figure 5.6: Transformation of the cancellation count series prior to modelling.

5.2.1 ARIMA Model

Following the transformation of the vendor cancellation series discussed previously, we proceed with the implementation of the ARIMA model as a baseline for time series forecasting. As outlined in the methodology, the ARIMA configuration was selected via an extensive grid search over various combinations of the (p, d, q) parameters, with the goal of minimizing the AIC while maintaining competitive predictive accuracy.

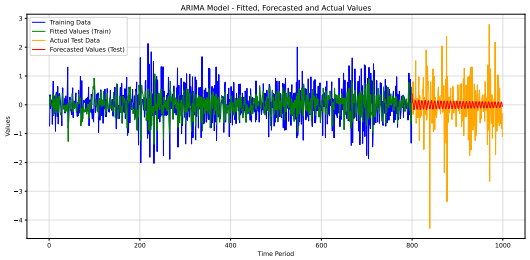
Table 5.1 summarizes the performance of the best ARIMA model selected for each of the three vendors under analysis.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Best Model	ARIMA(3, 0, 8)	ARIMA(0, 0, 3)	ARIMA(7, 0, 8)
MSE	0.7291	0.2822	0.8780
RMSE	0.8539	0.5312	0.9370
MAE	0.5944	0.3663	0.7365
AIC	1062.1396	697.6067	1724.0724

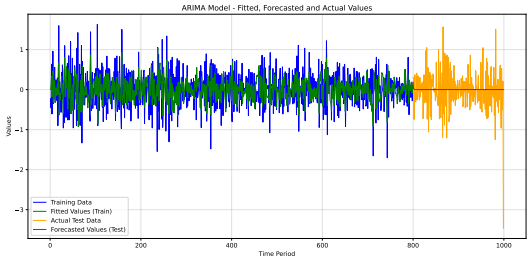
Table 5.1: ARIMA model performance metrics for the top offender vendors.

Figure 5.8 illustrates the models fit and forecast performance. The green curve shows the in-sample fitted values compared to the training data (blue), while the red curve represents the model's forecast over the test set, aligned against the actual test values in orange. In all the cases, the ARIMA models capture the overall level and general trend of the cancellation series but struggle to reflect its true dynamics. A strong seasonality pattern is present in the forecasts, which does not align with the actual fluctuations in the test data, especially around sharp increases or sudden drops. This suggests that while ARIMA provides a structured baseline, it lacks the flexibility needed to model the more irregular behaviour observed in

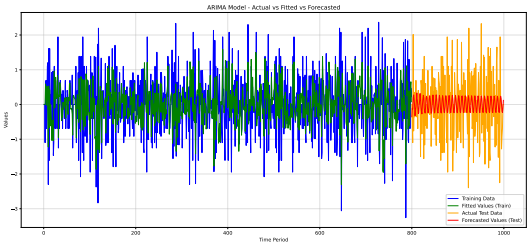
real cancellation patterns.



(a) ARIMA forecast for NA vendor



(b) ARIMA forecast for EU vendor



(c) ARIMA forecast for FE vendor

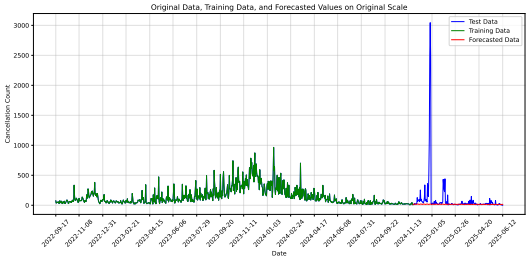
Figure 5.7: ARIMA model forecasts for vendor cancellations across the three top-offending vendors by region.

To better understand how the ARIMA forecasts behave in real-world terms, we transform the predictions back to the original scale of cancellation counts. The forecasted values for each of the three top vendors are shown in Figure 5.8. The models significantly underestimate both the level and variability of the actual values. The forecasts are overly smooth, with little responsiveness to the sharp increases and fluctuations present in the real data.

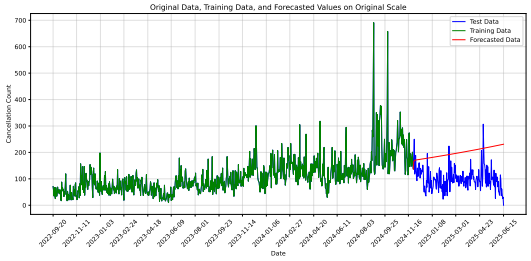
This qualitative observation is supported by the quantitative error metrics reported in Table 5.2.

Metric	**Vendor (NA)**	**Vendor (EU)**	**Vendor (FE)**
Best Model	ARIMA(3, 0, 8)	ARIMA(0, 0, 3)	ARIMA(7, 0, 8)
MSE	118608.56	12068.30	111.03
RMSE	344.40	109.86	10.54
MAE	79.32	101.73	5.58

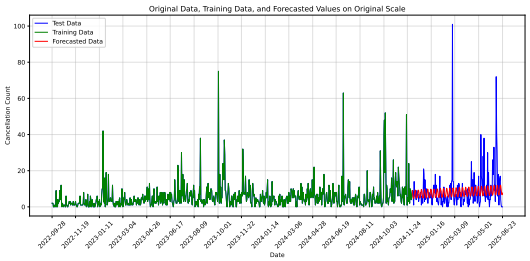
Table 5.2: ARIMA model performance metrics on the original scale for the top offender vendors.



(a) ARIMA forecast for NA vendor



(b) ARIMA forecast for EU vendor



(c) ARIMA forecast for FE vendor

Figure 5.8: ARIMA model forecasts for vendor cancellations across the three top-offending vendors by region, shown on the original scale.

To further assess the adequacy of the ARIMA model we examined its residuals. Figure 5.9 presents both the residuals over time and their rolling variance computed with a 20-period window. Rolling variance tracks how the dispersion of residuals evolves over time, providing insights into changes in the uncertainty of predictions. While the residuals appear mostly centred around zero, the rolling variance fluctuates significantly across the series, suggesting the presence of time-dependent variance. In a homoskedastic model, the rolling variance should remain relatively stable. However, in our case, the substantial variation indicates heteroskedastic behaviour. These results are consistent with our previous findings from the ACF of

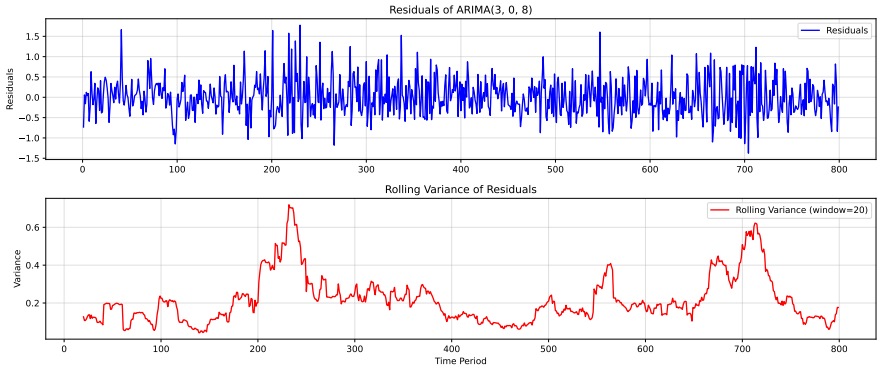


Figure 5.9: Residuals and 20-period rolling variance of the ARIMA(3, 0, 8) model for NA vendor.

the squared residuals (Figure 4.6) and the ARCH test discussed in the methodology, both of which point to the presence of ARCH effects. As a result, we conclude that a more flexible modelling approach is needed to accommodate the changing variance in the series. The next section introduces the ARMA-GARCH model, which explicitly models both the mean and variance dynamics.

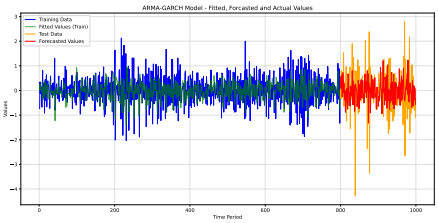
5.2.2 ARMA-GARCH Model

We now present the results obtained with the ARMA-GARCH model, which was implemented to address the limitations observed in the ARIMA approach, particularly its inability to capture time-dependent variance. As described in the methodology, the optimal configuration for each vendor was selected through a grid search over ARMA and GARCH parameter combinations, using the AIC as the primary selection criterion.

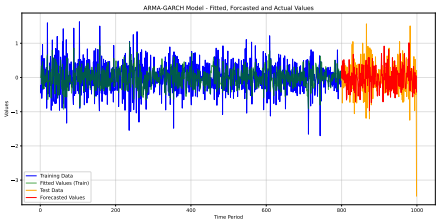
Table 5.3 summarizes the performance metrics of the best-fitting ARMA-GARCH models for the three vendors under analysis. Figure 5.10 illustrates the in-sample fits and out-of-sample forecasts of the ARMA-GARCH models on the differenced log-transformed series for the three vendors.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Best Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(1, 1)-GARCH(1, 1)
MSE	0.8813	0.3855	1.2750
RMSE	0.9388	0.6209	1.1292
MAE	0.6901	0.4737	0.8988
AIC	1072.7039	697.6537	1737.0598

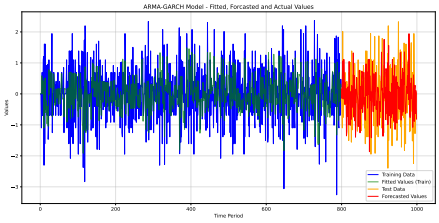
Table 5.3: ARMA-GARCH model performance metrics for the top offender vendors.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH for EU vendor

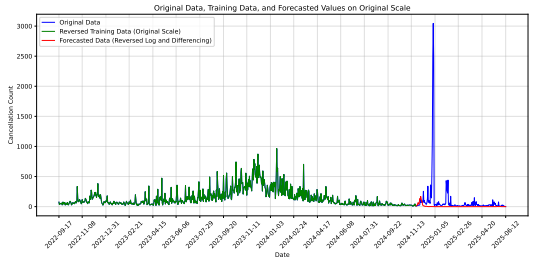


(c) ARMA-GARCH forecast for FE vendor

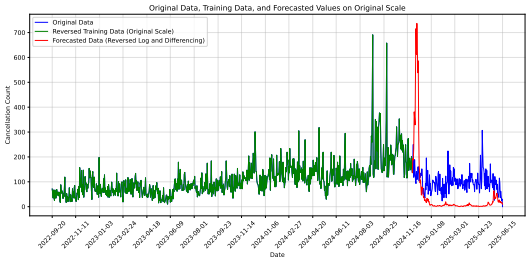
Figure 5.10: ARMA-GARCH model forecasts for vendor cancellations across the three top-offending vendors by region, shown on the differenced log-transformed scale.

Finally, we transform the ARMA-GARCH forecasts back to the original scale of cancellation counts for practical interpretability and error metric calculation. Figure 5.11 presents these forecasts alongside the actual cancellations. The ARMA-GARCH models demonstrate improved responsiveness to sharp increases and variability in cancellations compared to the ARIMA baseline, producing forecasts that are both smoother and more accurate in reflecting real-world behaviour, particularly in the first predicted values.

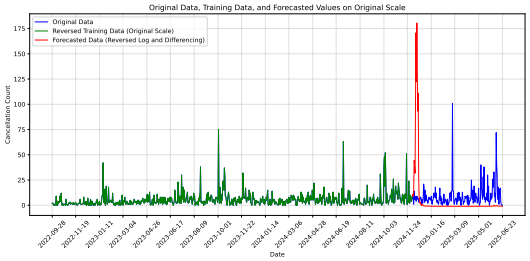
Table 5.4 reports the error metrics of the ARMA-GARCH models on the original cancellation count scale.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH for EU vendor



(c) ARMA-GARCH forecast for FE vendor

Figure 5.11: ARMA-GARCH model forecasts for vendor cancellations across the three top-offending vendors by region, shown on the original scale.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Best Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	121539.66	17965.50	782.93
RMSE	348.63	134.04	27.98
MAE	91.44	99.14	13.97

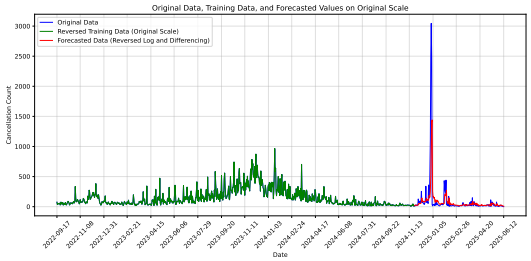
Table 5.4: ARMA-GARCH model performance metrics on the original scale for the top offender vendors.

Rolling Predictions

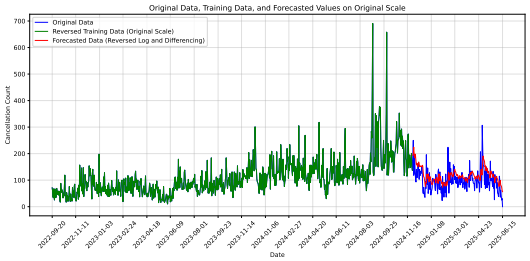
We now present the results of the rolling forecast evaluation using the ARMA-GARCH models. This framework was designed to more closely replicate a real-time forecasting environment by updating the model sequentially as new observations became available, as outlined in the methodology. Forecasts were produced for horizons of 1, 3, and 30 days across all three vendors, and performance was assessed on the original scale.

Figures 5.12, 5.13, and 5.14 illustrate the forecasts obtained at each rolling step for the three vendors, plotted against the actual observed cancellation counts.

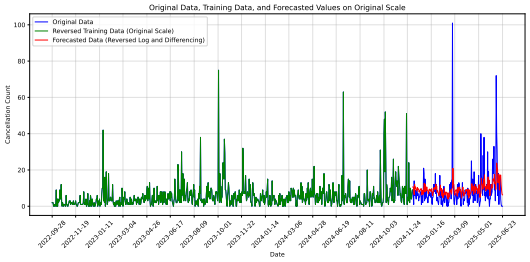
Tables 5.5, 5.6, and 5.7 report the prediction error metrics (MSE, RMSE, and



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH for EU vendor



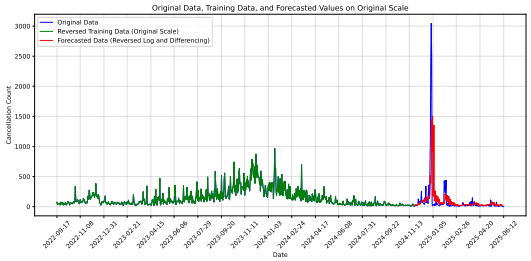
(c) ARMA-GARCH forecast for FE vendor

Figure 5.12: ARMA-GARCH model forecasts for vendor cancellations across the three top-offending vendors by region, using 1-day rolling predictions.

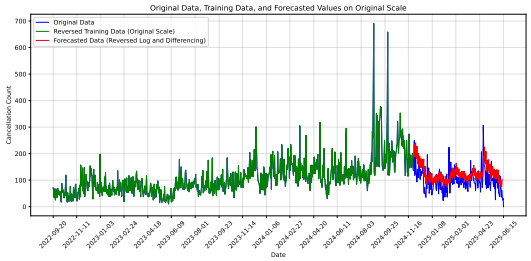
MAE) for each horizon. These tables provide a view of how forecast accuracy evolves with the length of the prediction window.

Metric	**Vendor (NA)**	**Vendor (EU)**	**Vendor (FE)**
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	64049.65	1816.07	106.31
RMSE	253.08	42.62	10.31
MAE	71.22	33.63	5.91

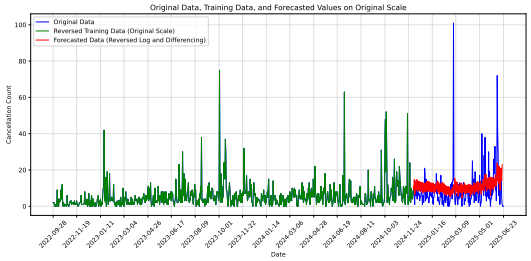
Table 5.5: ARMA-GARCH model performance metrics for the top offender vendors, using 1-day rolling predictions.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH for EU vendor



(c) ARMA-GARCH forecast for FE vendor

Figure 5.13: ARMA-GARCH model forecasts for vendor cancellations across the three top-offending vendors by region, using 3-day rolling predictions.

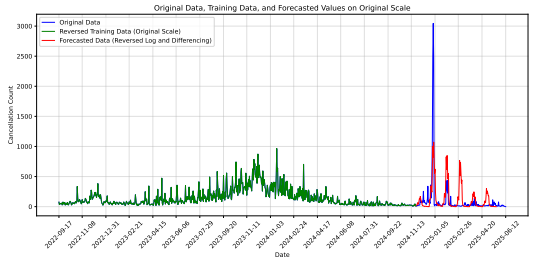
Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	88654.25	2050.57	104.95
RMSE	297.75	45.28	10.24
MAE	86.81	35.35	6.20

Table 5.6: ARMA-GARCH model performance metrics for the top offender vendors, using 3-day rolling predictions.

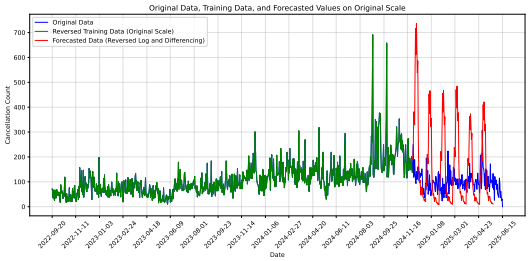
5.2.3 GRU Model

This section presents the performance of the GRU models trained as described in Section 4.2.4, evaluated across three forecast horizons: 1-day, 3-day, and 30-day ahead predictions. Figures 5.15, 5.16, and 5.17 display the model forecasts alongside actual vendor cancellation counts for each vendor and forecast horizon.

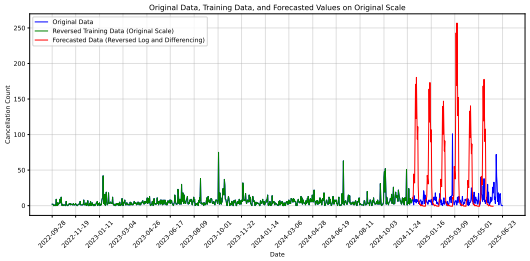
Corresponding performance results are summarized in Tables 5.8, 5.9, and 5.10. These tables report the evaluation metrics for each vendor and horizon, as well as the best hyperparameter settings identified through Bayesian optimization. This provides insight into the model configurations that yielded optimal forecasting accuracy for the different scenarios.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH for EU vendor



(c) ARMA-GARCH forecast for FE vendor

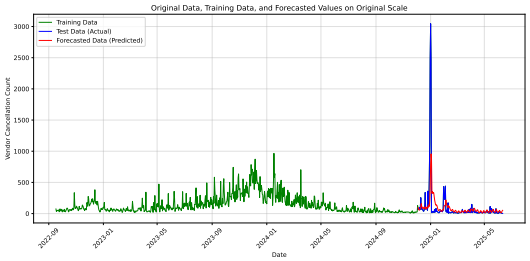
Figure 5.14: ARMA-GARCH model forecasts for vendor cancellations across the three top-offending vendors by region, using 30-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	83488.39	30319.34	4009.37
RMSE	288.94	174.13	63.32
MAE	126.04	123.26	36.61

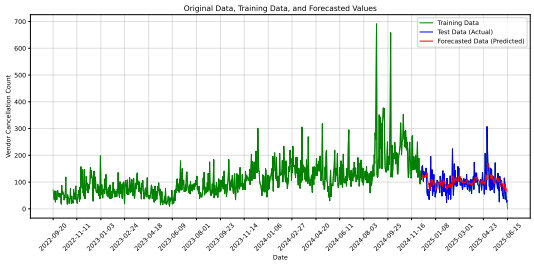
Table 5.7: ARMA-GARCH model performance metrics for the top offender vendors, using 30-day rolling predictions.

5.3 Scalability Results

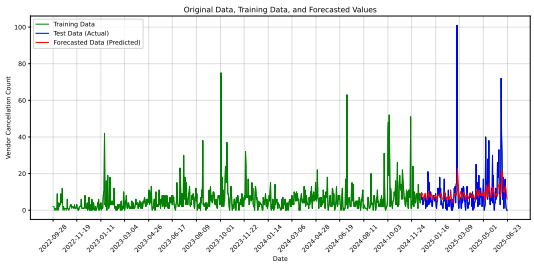
In this section, we present the performance of both ARMA-GARCH and GRU models when applied to a second, previously unseen vendor in each region. The models used here retain the hyperparameters optimized for the top-offending vendor in their respective regions. This setup allows us to evaluate the scalability of each modelling approach without retraining or tuning. Performance is measured again using 1-day, 3-day, and 30-day rolling forecasts. We report standard metrics (MSE, RMSE, MAE) in Tables 5.11, 5.12, 5.13, 5.14, 5.15, 5.16 and include the corresponding forecast plots in Figures 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, grouped



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor



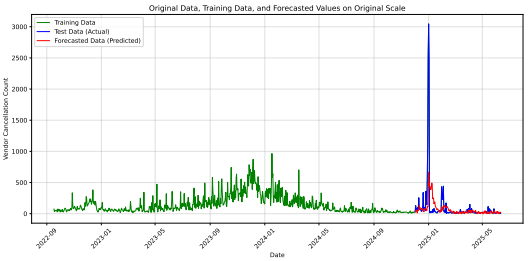
(c) GRU forecast for FE vendor

Figure 5.15: GRU model forecasts for vendor cancellations across the three top-offending vendors by region, using 1-day rolling predictions.

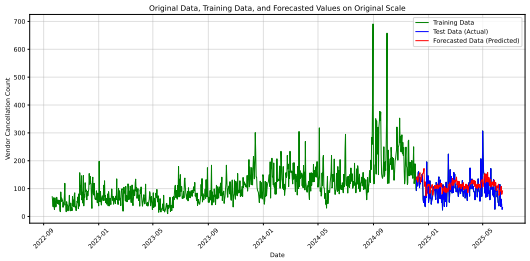
Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	1	1	1
units_per_layer	44	48	33
batch_size	8	8	32
MSE	78334.30	1387.91	108.99
RMSE	279.88	37.25	10.44
MAE	90.94	27.51	5.43

Table 5.8: GRU model performance metrics for the top offender vendors, using 1-day rolling predictions.

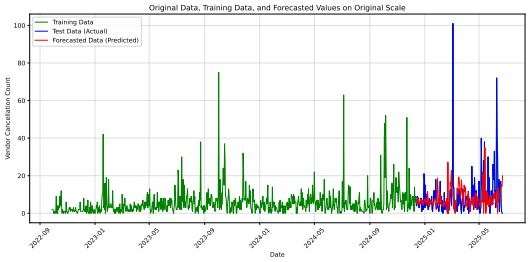
by model type and forecast horizon.



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor

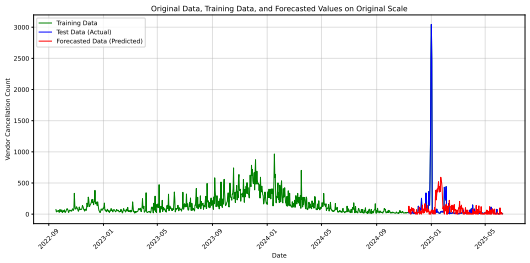


(c) GRU forecast for FE vendor

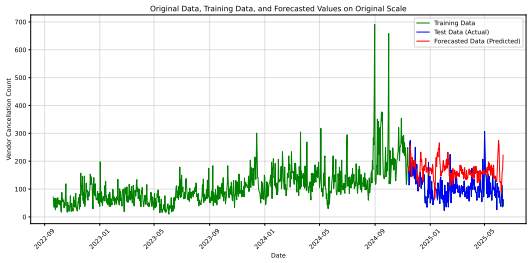
Figure 5.16: GRU model forecasts for vendor cancellations across the three top-offending vendors by region, using 3-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	1	3	3
units_per_layer	37	[46, 77, 91]	[63, 122, 95]
batch_size	8	32	32
MSE	96310.89	1817.05	145.49
RMSE	310.34	42.62	12.06
MAE	98.12	32.20	6.90

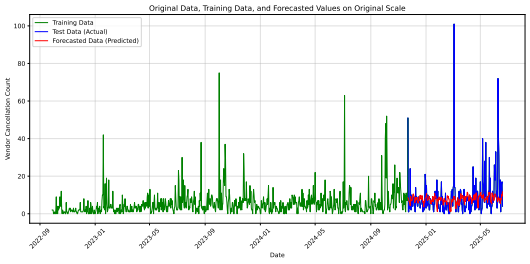
Table 5.9: GRU model performance metrics for the top offender vendors, using 3-day rolling predictions.



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor



(c) GRU forecast for FE vendor

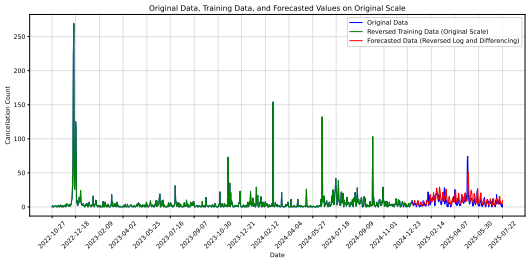
Figure 5.17: GRU model forecasts for vendor cancellations across the three top-offending vendors by region, using 30-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	2	3	1
units_per_layer	[82, 127]	[91, 62, 121]	115
batch_size	8	8	64
MSE	124354.58	6425.51	117.74
RMSE	352.64	80.15	10.85
MAE	121.30	68.77	5.77

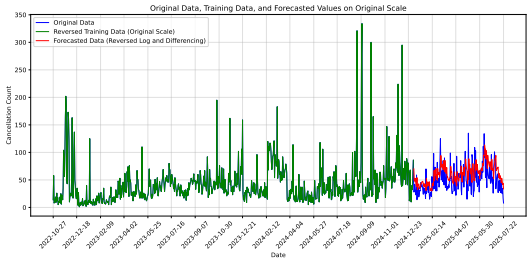
Table 5.10: GRU model performance metrics for the top offender vendors, using 30-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	66.26	652.65	209.68
RMSE	8.14	25.55	14.48
MAE	5.40	21.09	8.70

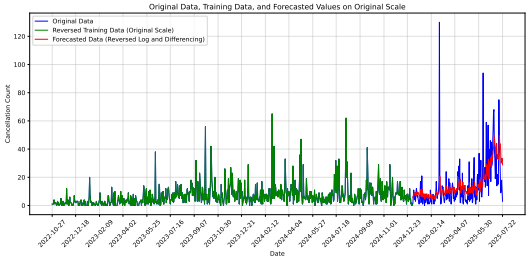
Table 5.11: ARMA-GARCH model performance metrics for unseen vendors, using 1-day rolling predictions.



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor



(c) GRU forecast for FE vendor

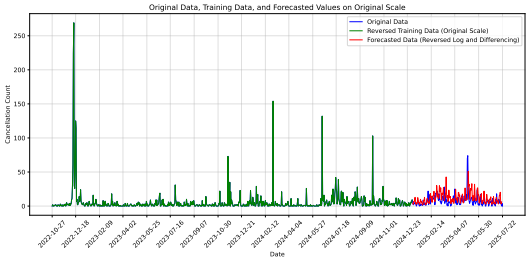
Figure 5.18: ARMA-GARCH model forecasts for vendor cancellations across three unseen vendors, using 1-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	105.04	886.64	232.33
RMSE	10.25	29.78	15.24
MAE	6.96	24.59	9.29

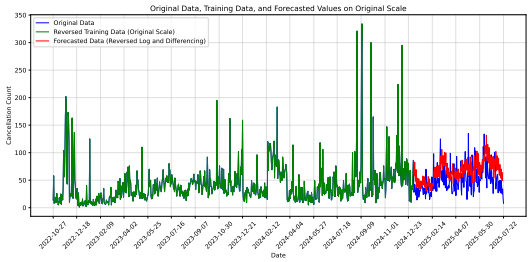
Table 5.12: ARMA-GARCH model performance metrics for unseen vendors, using 3-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
Model	ARMA(4, 4)-GARCH(1, 1)	ARMA(2, 1)-GARCH(1, 1)	ARMA(2, 2)-GARCH(1, 1)
MSE	10770.90	26046.62	5141.13
RMSE	103.78	161.39	71.70
MAE	49.89	105.67	43.58

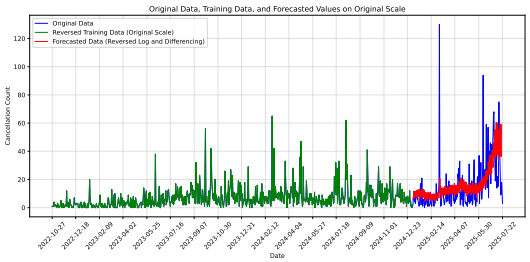
Table 5.13: ARMA-GARCH model performance metrics for unseen vendors, using 30-day rolling predictions.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH forecast for EU vendor

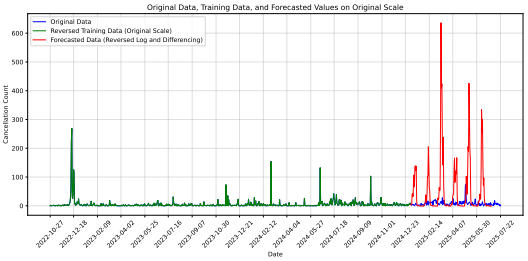


(c) ARMA-GARCH forecast for FE vendor

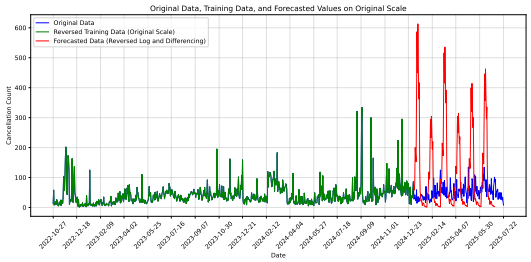
Figure 5.19: ARMA-GARCH model forecasts for vendor cancellations across three unseen vendors, using 3-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	1	1	1
units_per_layer	44	48	33
batch_size	8	8	32
MSE	66.43	598.32	227.78
RMSE	8.15	24.46	15.09
MAE	4.43	18.76	8.40

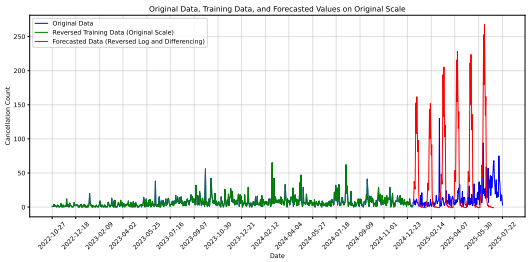
Table 5.14: GRU model performance metrics for unseen vendors, using 1-day rolling predictions.



(a) ARMA-GARCH forecast for NA vendor



(b) ARMA-GARCH forecast for EU vendor

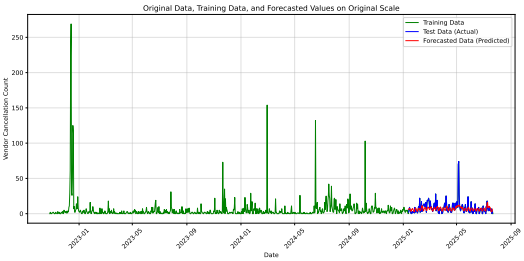


(c) ARMA-GARCH forecast for FE vendor

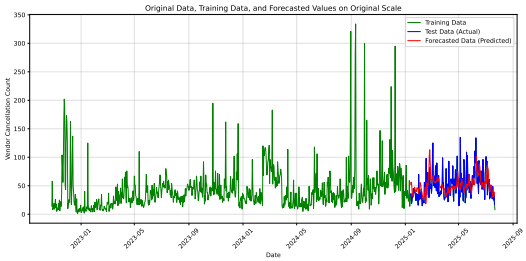
Figure 5.20: ARMA-GARCH model forecasts for vendor cancellations across three unseen vendors, using 30-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	1	3	3
units_per_layer	37	[46, 77, 91]	[63, 122, 95]
batch_size	8	32	32
MSE	71.74	531.66	289.70
RMSE	8.47	23.05	17.02
MAE	5.08	17.77	9.82

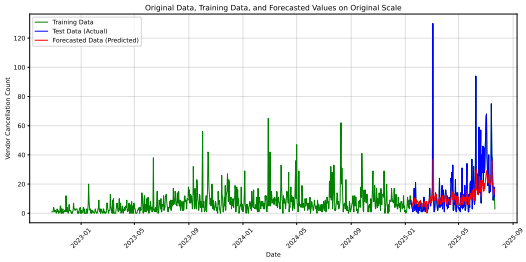
Table 5.15: GRU model performance metrics for unseen vendors, using 3-day rolling predictions.



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor

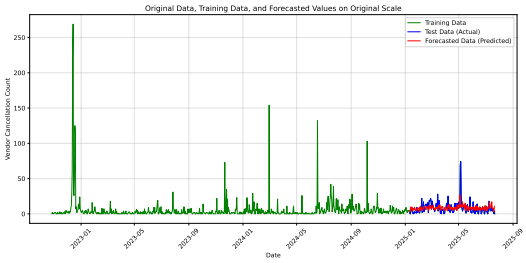


(c) GRU forecast for FE vendor

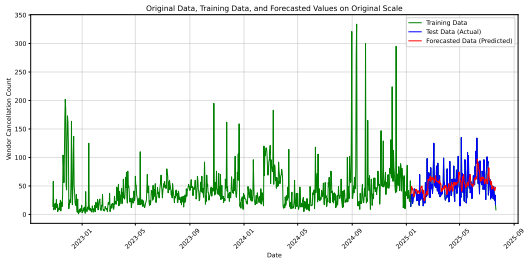
Figure 5.21: GRU model forecasts for vendor cancellations across three unseen vendors, using 1-day rolling predictions.

Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
n_layers	2	3	1
units_per_layer	[82, 127]	[91, 62, 121]	115
batch_size	8	8	64
MSE	93.57	1057.35	259.94
RMSE	9.67	32.52	16.12
MAE	6.58	24.34	9.11

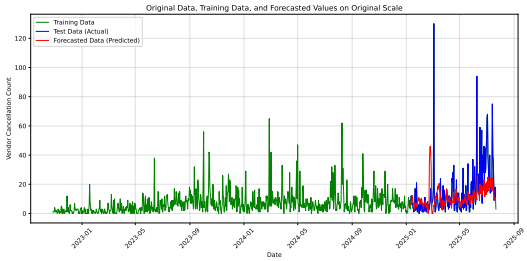
Table 5.16: GRU model performance metrics for unseen vendors, using 30-day rolling predictions.



(a) GRU forecast for NA vendor

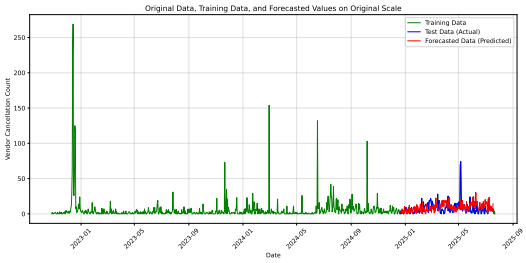


(b) GRU forecast for EU vendor

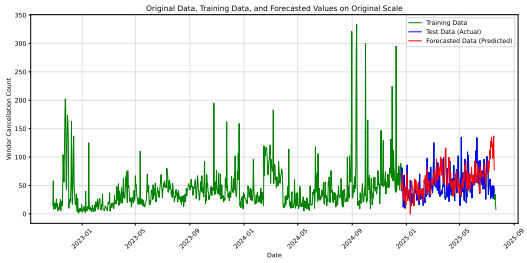


(c) GRU forecast for FE vendor

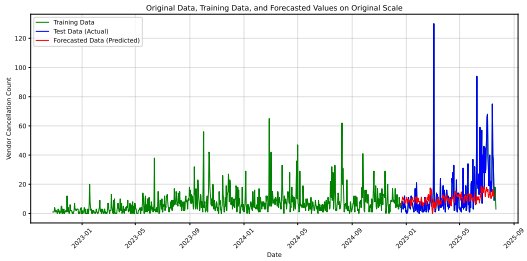
Figure 5.22: GRU model forecasts for vendor cancellations across three unseen vendors, using 3-day rolling predictions.



(a) GRU forecast for NA vendor



(b) GRU forecast for EU vendor



(c) GRU forecast for FE vendor

Figure 5.23: GRU model forecasts for vendor cancellations across three unseen vendors, using 30-day rolling predictions.

Chapter 6

Discussion

This chapter discusses the results obtained and their impact on vendor performance monitoring and forecasting at Amazon. It also highlights how different models can support decisions across short and long-term planning, depending on vendor behaviour and operational needs.

6.1 Vendor Health Dashboard

The implementation of the DF EI Health Dashboard represents a major step forward in Amazon's ability to monitor Electronic Integration performance at scale. The dashboard makes vendor reviews much easier by automatically gathering and sorting the data, cutting down on manual work. Tasks that previously required hundreds of hours annually to cover less then 40 vendors can now be performed in a matter of minutes, enabling broader and more frequent oversight across the global vendor network. This efficiency gain not only allows IOI teams to respond more quickly to integration issues, but also extends visibility to long-tail vendors that were previously unmonitored due to resource constraints.

Beyond improving coverage and responsiveness, the dashboard has also proven to adapt well in different operational settings. It has been actively adopted by EU IOI team in Spain, where much of the manual review process was historically concentrated, and has since been used in other key markets including the United States, Italy, Germany, Australia and Japan. These early deployments span multiple continents and demonstrate the tool's scalability and relevance across Amazon's global supply chain landscape. Figure 6.1 illustrates the current regional footprint of the dashboard's usage, reflecting its growing role in supporting integration health management worldwide.

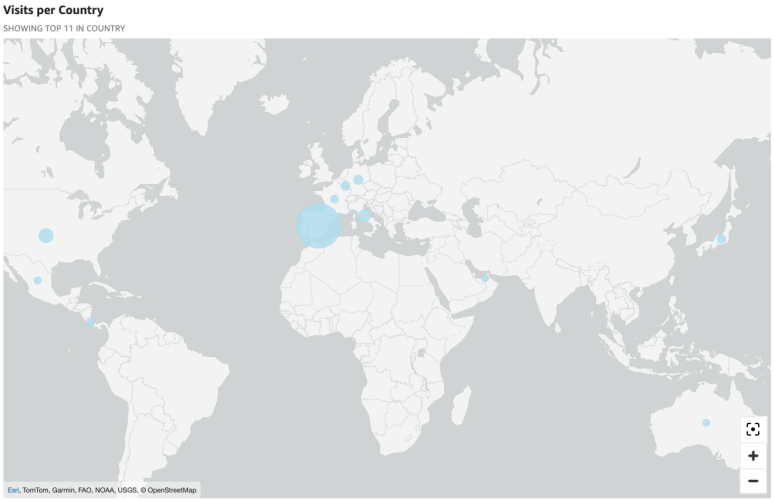


Figure 6.1: Geographic scope of Vendor Health Dashboard adoption.

6.2 Cancellation Forecasting

To provide a comprehensive comparison of forecasting performance, Table 6.1 summarizes the key error metrics for both the baseline ARIMA models and the ARMA-GARCH models, with static and rolling forecasts.

Starting with the baseline ARIMA models, we observe notable differences in forecasting accuracy across the vendors. However, it is important to note that each vendor's data presents distinct characteristics, such as differences in mean, variance, and overall scale, that affect error metrics. For example, the NA vendor has a substantially larger range and volume of orders and cancellations, leading to naturally higher error values (MSE: 118608.56, RMSE: 344.40, MAE: 79.32) compared to the FE vendor, which operates on a smaller scale and shows much lower errors (MSE: 111.03, RMSE: 10.54, MAE: 5.58). The EU vendor falls in between these two extremes with moderate error levels. Due to these inherent differences in data magnitude, direct comparisons of raw error metrics across vendors should be interpreted with caution, as higher errors may simply reflect larger underlying values rather than poorer model performance.

Overall, the ARIMA models perform poorly across all vendors. Their simplicity limits their ability to capture the complex dynamics and heteroskedasticity present in the data, resulting in forecasts that fail to explain much of the observed variance.

The ARMA-GARCH models introduce volatility modelling to capture conditional heteroskedasticity in the data. This added complexity influences forecast performance differently across vendors and horizons. In the static setting, ARMA-GARCH

Model Category	Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
ARIMA (Baseline)				
	Model	ARIMA(3,0,8)	ARIMA(0,0,3)	ARIMA(7,0,8)
	MSE	118608.56	12068.30	111.03
	RMSE	344.40	109.86	10.54
	MAE	79.32	101.73	5.58
ARMA-GARCH				
	Model	ARMA(4,4)-GARCH(1,1)	ARMA(2,1)-GARCH(1,1)	ARMA(2,2)-GARCH(1,1)
Static forecast				
	MSE	121539.66	17965.50	782.93
	RMSE	348.63	134.04	27.98
	MAE	91.44	99.14	13.97
1-day rolling forecast				
	MSE	64049.65	1816.07	106.31
	RMSE	253.08	42.62	10.31
	MAE	71.22	33.63	5.91
3-day rolling forecast				
	MSE	88654.25	2050.57	104.95
	RMSE	297.75	45.28	10.24
	MAE	86.81	35.35	6.20
30-day rolling forecast				
	MSE	83488.39	30319.34	4009.37
	RMSE	288.94	174.13	63.32
	MAE	126.04	123.26	36.61

Table 6.1: Original-scale performance metrics for all ARIMA and ARMA-GARCH models across the three top-offending vendors. ARIMA serves as the baseline, while ARMA-GARCH results are presented for static and rolling forecast horizons.

errors are generally comparable to or slightly higher than those of the simpler ARIMA models, especially for the NA and EU vendors. This indicates that without frequent re-estimation, the benefits of explicitly modelling volatility remain limited.

A key characteristic of ARMA-GARCH processes is their mean-reverting nature. In this application, both the mean and the variance dynamics are explicitly modelled. As a result, these models tend to perform reasonably well for short-term predictions, particularly the first forecast step, where the effects of recent volatility are still present. However, as the forecast horizon extends, the predictions tend to revert to the process mean, gradually losing their ability to reflect recent fluctuations. This tendency limits their usefulness in capturing longer-range dynamics.

However, the advantage of ARMA-GARCH becomes evident in rolling forecasts, where the model is updated regularly with new information. The 1-day rolling forecasts show a pronounced reduction in error metrics across all vendors. For instance, the NA vendor's MSE decreases from 121539.66 (static) to 64049.65 (1-day rolling), nearly halving the error. Similar improvements are observed for the EU vendor, where MSE drops from 17965.50 to 1816.07, and for the FE vendor, from 782.93 to 106.31.

Extending the forecast horizon to 3 days, errors increase slightly compared to the 1-day rolling forecasts but remain well below static forecast errors. This indicates that ARMA-GARCH models maintain reasonable predictive power with moderate horizon extension.

At the 30-day rolling forecast horizon, prediction errors rise substantially, particularly for the NA and EU vendors (MSE of 83488.39 and 30319.34, respectively), emphasising the increased uncertainty and difficulty in modelling longer-term forecasts under volatile conditions. While the FE vendor also shows increased error (MSE: 4009.37), the impact is more moderate, consistent with its relatively stable and less erratic cancellation patterns.

Visual inspection of the forecasted values reveals a notable shortcoming in the model behaviour at this horizon. Specifically, the ARMA-GARCH models appear to overestimate the variance of the series, producing periodic spikes that do not align with the actual test data. This pattern suggests that the models may be reacting too strongly to recent volatility, projecting it forward in a way that inflates uncertainty rather than capturing realistic dynamics.

These limitations, especially the difficulty in capturing variance and maintaining accurate long-range forecasts, highlighted the need to supplement the statistical models with a more flexible, data-driven method. To that end, we summarize the performance of the GRU models in Table 6.2, which presents key error metrics and model configurations across rolling horizons for the top-offending vendors.

The GRU models demonstrate a clear improvement in forecasting performance over the traditional statistical baselines, particularly as the forecast horizon increases.

For the 1-day rolling forecast, GRU models achieve strong performance with relatively simple architectures. The FE vendor exhibits low RMSE and MAE (10.44 and 5.43, respectively), consistent with the stable patterns observed in its cancellation data. The EU and NA vendors, while operating on more volatile or larger-scale series, also benefit from the GRU's ability to model short-term nonlinear dependencies effectively.

At the 3-day horizon, increased model complexity becomes noticeable, particularly for the EU and FE vendors, which both converge to three-layer architectures. The models maintain good performance, suggesting the GRU's capacity to adapt to medium-range temporal dynamics. Although errors increase modestly compared to the 1-day case, they remain well below the levels observed in traditional models for comparable horizons.

At the 30-day rolling forecast horizon, the GRU model demonstrates a marked improvement over the ARMA-GARCH approach for the European and Far East vendors, with notably lower RMSE and MAE values. This indicates that the GRU is better able to capture the temporal dynamics in these less variable and more

Forecast Horizon	Metric	Vendor (NA)	Vendor (EU)	Vendor (FE)
1-day rolling forecast				
	n_layers	1	1	1
	units_per_layer	44	48	33
	batch_size	8	8	32
	MSE	78334.30	1387.91	108.99
	RMSE	279.88	37.25	10.44
	MAE	90.94	27.51	5.43
3-day rolling forecast				
	n_layers	1	3	3
	units_per_layer	37	[46, 77, 91]	[63, 122, 95]
	batch_size	8	32	32
	MSE	96310.89	1817.05	145.49
	RMSE	310.34	42.62	12.06
	MAE	98.12	32.20	6.90
30-day rolling forecast				
	n_layers	2	3	1
	units_per_layer	[82, 127]	[91, 62, 121]	115
	batch_size	8	8	64
	MSE	124354.58	6425.51	117.74
	RMSE	352.64	80.15	10.85
	MAE	121.30	68.77	5.77

Table 6.2: Summary of GRU model architecture and performance metrics for the top offender vendors, across 1-day, 3-day, and 30-day rolling forecast horizons.

stable series. For the North America vendor, however, the forecasting challenge remains greater due to its higher volatility, and the GRU errors are somewhat larger compared to ARMA-GARCH. Beyond the quantitative metrics, visual inspection of the GRU forecasts reveals smoother and more realistic prediction trajectories across all vendors. Unlike the ARMA-GARCH forecasts, which occasionally exhibited unrealistic periodic spikes especially at longer horizons, the GRU outputs are more stable and coherent.

Table 6.3 reveals that model performance varies significantly by vendor and forecast horizon. For the NA vendor, characterized by high volatility and scale, ARMA-GARCH models consistently achieve lower error metrics than GRU models across all rolling forecast horizons, reflecting their relative strength in capturing short to medium-term volatility dynamics in this dataset. In contrast, the EU vendor benefits notably from GRU modelling, with GRU forecasts outperforming ARMA-GARCH across all rolling horizons, particularly at longer horizons where GRU's MSE is substantially lower. The FE vendor presents a mixed picture: ARMA-GARCH models yield slightly better accuracy at shorter horizons (1-day

and 3-day rolling), while GRU models significantly outperform ARMA-GARCH at the 30-day rolling forecast horizon, suggesting GRU's ability to better model longer-term trends in this more stable vendor data. Across all vendors, static forecasts perform worse than rolling forecasts, highlighting the value of frequent model updates, except for the 30-day forecasts. This can be attributed to the fact that while two out of three of the static models predict one spike in variance which does not realize, the rest of the prediction hover around the mean. While for the 30 day models the model contentiously overshoots the realized values.

Overall, these results suggest that while ARMA-GARCH is more suitable for volatile, large-scale data like NA, GRU models offer clear advantages for vendors with more stable patterns, especially over longer forecast horizons. Visual inspection corroborates these findings, as GRU forecasts tend to produce smoother, more realistic trajectories compared to the sometimes volatile ARMA-GARCH outputs.

6.2.1 Scalability Assessment

The scalability evaluation compares ARMA-GARCH and GRU models trained on the top offenders in each region and then directly applied, without retraining, to a second, previously unseen vendor in the same region. Tables 5.11, 5.16 report forecasting performance across 1-day, 3-day, and 30-day rolling windows.

For 1-day rolling forecasts, both model families performed comparably across most regions. GRUs slightly outperformed ARMA-GARCH in NA (MAE: 4.43 vs. 5.40) and EU (18.76 vs. 21.09), while the results were similar in the FE (8.40 vs. 8.70). This suggests that the GRU model generalizes well to similar vendor time series over short horizons, possibly due to its ability to capture nonlinear temporal dependencies.

At the 3-day horizon, GRU models again demonstrated slightly lower MAE values across all regions compared to ARMA-GARCH. Notably, in the EU region, GRU reduced the MAE to 17.77 from 24.59, indicating better adaptability despite a more complex architecture (i.e., three hidden layers).

However, for long-term forecasts (30-day rolling), both models experienced significant error growth, as expected. ARMA-GARCH performance degraded more sharply, e.g. MAE reached 105.67 in the EU, whereas GRU maintained comparatively lower errors (MAE: 24.34). This indicates stronger robustness of GRUs to temporal drift in longer horizons, likely due to their recurrent memory capabilities.

Overall, both models are able to transfer reasonably well to unseen vendors without ad hoc modelling. GRUs consistently achieved slightly better or equivalent accuracy, particularly at longer horizons and in more variable regional contexts. However, ARMA-GARCH maintained competitive performance with far simpler architectures. Importantly, the accuracy levels observed on the second set of

Model	Forecast	Metric	NA	EU	FE
ARIMA (Baseline)					
	Static	MSE	118,608.56	12,068.30	111.03
		RMSE	344.40	109.86	10.54
		MAE	79.32	101.73	5.58
ARMA-GARCH					
	Static	MSE	121,539.66	17,965.50	782.93
		RMSE	348.63	134.04	27.98
		MAE	91.44	99.14	13.97
	1-day rolling	MSE	64,049.65	1,816.07	106.31
		RMSE	253.08	42.62	10.31
		MAE	71.22	33.63	5.91
	3-day rolling	MSE	88,654.25	2,050.57	104.95
		RMSE	297.75	45.28	10.24
		MAE	86.81	35.35	6.20
	30-day rolling	MSE	83,488.39	30,319.34	4,009.37
		RMSE	288.94	174.13	63.32
		MAE	126.04	123.26	36.61
GRU					
	1-day rolling	MSE	78,334.30	1,387.91	108.99
		RMSE	279.88	37.25	10.44
		MAE	90.94	27.51	5.43
	3-day rolling	MSE	96,310.89	1,817.05	145.49
		RMSE	310.34	42.62	12.06
		MAE	98.12	32.20	6.90
	30-day rolling	MSE	124,354.58	6,425.51	117.74
		RMSE	352.64	80.15	10.85
		MAE	121.30	68.77	5.77

Table 6.3: Comparison of key performance metrics (MSE, RMSE, MAE) across ARIMA, ARMA-GARCH, and GRU models for NA, EU, and FE vendors. Rolling forecast horizons are shown where applicable.

vendors remained comparable to those on the original vendors, suggesting that both approaches are scalable across similar vendor time series. This supports the feasibility of reusing trained models regionally without per-vendor hyperparameter tuning.

Building on these modelling insights, the Amazon IOI team can apply customized

forecasting methods to better manage vendor performance at scale. For vendors exhibiting higher volatility or error rates, integrating ARMA-GARCH models into daily monitoring processes can improve short-term accuracy and enable more responsive interventions. These models are especially useful when the focus is on capturing rapid fluctuations or variance patterns, such as during high-pressure periods like *Prime Week*[1], when operations teams need precise, day-level predictions to minimize disruptions.

In contrast, for more stable vendors, GRU-based forecasts offer a better fit. These models are good at identifying underlying trends and maintaining smoother, more consistent predictions over longer horizons. Their ability to avoid spikes or anomalous behaviour makes them well suited for use cases like weekly or monthly planning, where business stakeholders are more interested in average cancellation rates and long-term performance.

Importantly, the choice between models doesn't need to be static. Forecasting needs vary with the time frame and operational context. GRUs generally perform better overall, both in terms of accuracy and the visual quality of the forecasts, and should be the default for strategic, forward-looking assessments. However, ARMA-GARCH remains a practical option for short-term use cases that demand interpretability and quick updates, particularly in volatile conditions.

[1]Prime Week is an annual sales event created by Amazon to celebrate its anniversary and offer exclusive deals to Amazon Prime members.

Chapter 7

Conclusion

This thesis focused on developing systematic methods to monitor and forecast vendor performance and electronic integration quality at scale within Amazon's supply chain ecosystem. By building and deploying the Vendor Health Dashboard, we automated and scaled vendor health monitoring to cover thousands of global partners, reducing hours of manual effort and expanding oversight to long-tail vendors who were previously left out from regular reviews. This approach demonstrates how vendor integration health can be managed effectively using a centralized, scalable tool that's accessible throughout Amazon's network

On the forecasting side, we conducted a comparative analysis of classical time series models, including ARIMA and ARMA-GARCH, alongside a neural network approach based on Gated Recurrent Units. Our results showed that ARMA-GARCH models perform well for highly volatile vendors in short-term forecasting, while GRUs consistently outperform at longer forecast horizons, particularly for vendors with more stable cancellation patterns. This demonstrates the value of hybrid forecasting strategies that adapt model choice and forecast frequency to vendor-specific behaviours, improving accuracy and robustness. Furthermore, these predictive models proved capable of anticipating vendor cancellations sufficiently in advance, allowing more proactive interventions to address emerging issues before they escalate.

The scalability of the forecasting models was also evaluated by testing their ability to generalize to unseen vendors without additional fine-tuning of the parameters. Results showed that both ARMA-GARCH and GRU models maintained strong performance across new vendors from different regions, suggesting that a region-specific model can be reused in similar operational contexts. This supports the feasibility of deploying forecasting at scale, where building custom models for each vendor may be impractical.

Looking ahead, there are several promising directions for future work. The dashboard could be extended with real-time alerting systems that automatically notify the IOI team when critical thresholds are exceeded, such as a sudden spike in cancellation rates or prolonged delays in order fulfillment. Furthermore, incorporating methods to assess the effects of specific interventions, such as the implementation of a new API version or a change in EI configuration, could help identify whether these actions are driving observed changes in vendor performance.

Beyond methodological improvements, future iterations of the dashboard could incorporate new metrics to deepen the evaluation of vendor integration. Two specific additions have been identified as valuable for further improving the dashboard. The first is Shipping Label Processing Delays, which would track the time elapsed between the submission of a label request and the receipt of the label file by the vendor. This metric is especially relevant for vendors that rely on Amazon-provided shipping labels through EDI or API integrations. In these cases, any delay in the transmission or processing of label information can introduce bottlenecks in order preparation and shipping. Monitoring this delay would allow Amazon to assess how efficiently the label exchange process is functioning and to identify vendors experiencing latency in this step of the fulfillment process. The second proposed metric is API Certification Status, which monitors the validity period of the vendor's API credentials, specifically, the Login with Amazon (LWA) tokens required for secure system-to-system communication. Rather than serving as a direct performance metric, this functions as an early-warning indicator, alerting the IOI team when a vendor's certification is approaching expiration. Adding this flag to the dashboard would help the team reach out to vendors in advance to renew their credentials, reducing the risk of last-minute disruptions caused by expired credentials and ensuring continuity in data exchange.

Regarding forecasting, adding more data, such as inventory levels, or details about each vendor, could improve both the accuracy and robustness of the models. Moreover, when forecasting models are deployed at scale within a company, it becomes crucial to implement automated pipelines for continuous model evaluation, retraining, and monitoring. Such systems could monitor model performance metrics in real time, trigger retraining when performance drops below certain thresholds, and deploy updated models without manual intervention.

Bibliography

[1] Rafay Ishfaq and Uzma Raja. «Evaluation of order fulfillment options in retail supply chains». In: *Decision Sciences* 49.3 (2018), pp. 487–521 (cit. on p. 1).

[2] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970 (cit. on pp. 4, 5, 11).

[3] Tim Bollerslev. «On the correlation structure for the generalized autoregressive conditional heteroskedastic process». In: *Journal of time series analysis* 9.2 (1988), pp. 121–131 (cit. on pp. 4, 5, 12).

[4] Rahul Dey and Fathi M Salem. «Gate-variants of gated recurrent unit (GRU) neural networks». In: *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE. 2017, pp. 1597–1600 (cit. on pp. 4, 6).

[5] Larry R Medsker, Lakhmi Jain, et al. «Recurrent neural networks». In: *Design and applications* 5.64-67 (2001), p. 2 (cit. on p. 6).

[6] Sepp Hochreiter and Jürgen Schmidhuber. «Long short-term memory». In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 6).

[7] Jingyi Wang, Li-Chun Yu, K Robert Lai, and Xuejie Zhang. «OGRU: An optimized gated recurrent unit neural network». In: *Pattern Recognition* 88 (2019), pp. 144–155 (cit. on p. 7).

[8] Guoqiang Jin, Tianyi Zhu, Muhammad Waqar Akram, Yi Jin, and Changan Zhu. «An adaptive anti-noise neural network for bearing fault diagnosis under noise and varying load conditions». In: *Ieee Access* 8 (2020), pp. 74793–74807 (cit. on p. 7).

[9] James Bergstra and Yoshua Bengio. «Random search for hyper-parameter optimization». In: *The journal of machine learning research* 13.1 (2012), pp. 281–305 (cit. on p. 7).

[10] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. «Practical bayesian optimization of machine learning algorithms». In: *Advances in neural information processing systems* 25 (2012) (cit. on p. 7).

[11] Peter I. Frazier. *A Tutorial on Bayesian Optimization*. 2018. arXiv: 1807.02811 [stat.ML] (cit. on p. 8).

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. «An introduction to statistical learning». In: (2009) (cit. on p. 8).

[13] Hirotugu Akaike. «Factor analysis and AIC». In: *Psychometrika* 52.3 (1987), pp. 317–332 (cit. on p. 9).

[14] Said E Said and David A Dickey. «Testing for unit roots in autoregressive-moving average models of unknown order». In: *Biometrika* 71.3 (1984), pp. 599–607 (cit. on p. 9).

[15] Robert F Engle. «Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation». In: *Econometrica: Journal of the Econometric Society* (1982), pp. 987–1007 (cit. on pp. 10, 12).

[16] Luis Aburto and Richard Weber. «Improved supply chain management based on hybrid demand forecasts». In: *Applied Soft Computing* 7.1 (2007), pp. 136–144 (cit. on p. 11).

[17] Ping-Feng Pai and Chih-Sheng Lin. «A hybrid ARIMA and support vector machines model in stock price forecasting». In: *Omega* 33.6 (2005), pp. 497–505 (cit. on p. 11).

[18] Volkan Ş Ediger and Sertac Akar. «ARIMA forecasting of primary energy demand by fuel in Turkey». In: *Energy Policy* 35.3 (2007), pp. 1701–1708 (cit. on p. 11).

[19] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. «ARIMA models to predict next-day electricity prices». In: *IEEE Transactions on Power Systems* 18.3 (2003), pp. 1014–1020 (cit. on p. 11).

[20] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. «An empirical comparison of machine learning models for time series forecasting». In: *Econometric Reviews* 29.5-6 (2010), pp. 594–621 (cit. on p. 11).

[21] G Peter Zhang. «Time series forecasting using a hybrid ARIMA and neural network model». In: *Neurocomputing* 50 (2003), pp. 159–175 (cit. on p. 12).

[22] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018 (cit. on p. 12).

[23] K. Miramadi and H. Jiang. *Predicting Intraday Trading Volumes on The Swedish Market Using Deep Learning*. 2025 (cit. on p. 12).

[24] Chew Weng Chong, Maheran I. Ahmad, and Mazlina Y. Abdullah. «Performance of GARCH models in forecasting stock market volatility». In: *Journal of Forecasting* 18.5 (1999), pp. 333–343 (cit. on p. 12).

[25] Emilio Barucci and Roberto Renò. «On measuring volatility and the GARCH forecasting performance». In: *Journal of International Financial Markets, Institutions and Money* 12.3 (July 2002), pp. 183–200 (cit. on p. 12).

[26] Juri Marcucci. «Forecasting Stock Market Volatility with Regime-Switching GARCH Models». In: *Studies in Nonlinear Dynamics & Econometrics* 9.4 (2005) (cit. on p. 12).

[27] R.C. Garcia, J. Contreras, M. van Akkeren, and J.B.C. Garcia. «A GARCH forecasting model to predict day-ahead electricity prices». In: *IEEE Transactions on Power Systems* 20.2 (2005), pp. 867–874 (cit. on p. 12).

[28] Yue-Jun Zhang, Ting Yao, Ling-Yun He, and Ronald Ripple. «Volatility forecasting of crude oil market: Can the regime switching GARCH model beat the single-regime GARCH models?» In: *International Review of Economics & Finance* 59 (Jan. 2019), pp. 302–317 (cit. on p. 12).

[29] Peter R Hansen and Asger Lunde. «A forecast comparison of volatility models: does anything beat a GARCH (1, 1)?» In: *Journal of Applied Econometrics* 20.7 (2005), pp. 873–889 (cit. on p. 12).

[30] Hong Thom Pham and Bo-Suk Yang. «Estimation and forecasting of machine health condition using ARMA/GARCH model». In: *Mechanical Systems and Signal Processing* 24.2 (Feb. 2010), pp. 546–558 (cit. on p. 12).

[31] Shoumen Datta, Don P. Graham, Nikhil Sagar, Pat Doody, Reuben Slone, and Olli-Pekka Hilmola. «Forecasting and Risk Analysis in Supply Chain Management: GARCH Proof of Concept». In: *Managing Supply Chain Risk and Vulnerability*. Ed. by Teresa Wu and Jennifer Blackhurst. London: Springer London, 2009, pp. 187–203. ISBN: 978-1-84882-633-5 (cit. on p. 12).

[32] Bradley T. Ewing and Mark A. Thompson. «Industrial production, volatility, and the supply chain». In: *International Journal of Production Economics* 115.2 (Oct. 2008), pp. 553–558 (cit. on p. 12).

[33] Jianxin Chen, Xiande Zhao, and Zhongying Shen. «Supply chain risk assessment and control of bullwhip effect based on GARCH model». In: *International Journal of Production Research* 51.10 (2013), pp. 3115–3128 (cit. on p. 12).

[34] Xun Wang, Shuai Hao, and Liang Yao. «Forecasting extreme risk in multi-level supply chains: A GARCH-EVT approach». In: *International Journal of Production Research* 54.22 (2016), pp. 6634–6652 (cit. on p. 12).

[35] Rafal Weron and Adam Misiorek. «Forecasting wholesale electricity prices: A review of time series models». In: *HSBC's Guide to Cash and Treasury Management in Asia Pacific* (2008), pp. 744–763 (cit. on p. 12).

[36] Zheng Tan, Jinliang Zhang, Jianhui Wang, and Jun Xu. «Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models». In: *Applied Energy* 87.11 (2010), pp. 3606–3610 (cit. on p. 13).

[37] Luc Bauwens, Sébastien Laurent, and Jeroen VK Rombouts. «Multivariate GARCH models: a survey». In: *Journal of Applied Econometrics* 21.1 (2006), pp. 79–109 (cit. on p. 13).

[38] Annastiina Silvennoinen and Timo Teräsvirta. «Multivariate GARCH models». In: *Handbook of Financial Time Series* (2009), pp. 201–229 (cit. on p. 13).

[39] Richard T Baillie, Tim Bollerslev, and Hans Ole Mikkelsen. «Fractionally integrated generalized autoregressive conditional heteroskedasticity». In: *Journal of Econometrics* 74.1 (1996), pp. 3–30 (cit. on p. 13).

[40] Daniel B Nelson. «Conditional heteroskedasticity in asset returns: A new approach». In: *Econometrica: Journal of the Econometric Society* (1991), pp. 347–370 (cit. on p. 13).

[41] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: `1409.1259 [cs.CL]` (cit. on p. 13).

[42] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: `1412.3555 [cs.NE]` (cit. on p. 13).

[43] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. «An empirical exploration of recurrent network architectures». In: *International Conference on Machine Learning*. 2015, pp. 2342–2350 (cit. on p. 13).

[44] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. «Light gated recurrent units for speech recognition». In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2018, pp. 37–43 (cit. on p. 13).

[45] Guizhu Shen, Qingping Tan, Haoyu Zhang, Ping Zeng, and Jianjun Xu. «Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions». In: *Procedia Computer Science* 131 (2018), pp. 895–903 (cit. on p. 13).

[46] Thomas Fischer and Christopher Krauss. «Deep learning with long short-term memory networks for financial market predictions». In: *European Journal of Operational Research* 270.2 (2018), pp. 654–669 (cit. on p. 13).

[47] Hyeong Kyu Kim and Kyungsik Han. «Predicting the direction of US stock prices using effective transfer entropy and machine learning techniques». In: *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2019, pp. 259–269 (cit. on p. 14).

[48] Wenjuan Shu, Fanping Zeng, Zhen Ling, Junyi Liu, Tingting Lu, and Guozhu Chen. «Resource Demand Prediction of Cloud Workloads Using an Attention-based GRU Model». In: *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. 2021, pp. 428–437 (cit. on p. 14).

[49] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. «Independently recurrent neural network (indrnn): Building a longer and deeper rnn». In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 5457–5466 (cit. on p. 14).

[50] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. «Dual-stage attention-based recurrent neural network for time series prediction». In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2017, pp. 2627–2633 (cit. on p. 14).

[51] Md Abrar Jahin, Asef Shahriar, and Md Al Amin. *MCDFN: Supply Chain Demand Forecasting via an Explainable Multi-Channel Data Fusion Network Model*. 2025. arXiv: `2405.15598 [cs.LG]` (cit. on p. 14).

[52] W.A. Roshan S. Jayasekara, P.T. Ranil S. Sugathadasa, Oshadhi Herath, and Niles Perera. «Multivariate Sales Forecasting Using Gated Recurrent Unit Network Model». In: *International Journal of Supply and Operations Management* 11.4 (2024), pp. 390–416 (cit. on p. 14).

[53] Jiseong Noh, Hyun-Ji Park, Jong Soo Kim, and Seung-June Hwang. «Gated Recurrent Unit with Genetic Algorithm for Product Demand Forecasting in Supply Chain Management». In: *Mathematics* 8.4 (2020), p. 565 (cit. on p. 14).

[54] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. «Recurrent neural networks for multivariate time series with missing values». In: *Scientific Reports* 8.1 (2018), pp. 1–12 (cit. on p. 14).

[55] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. «NAOMI: Non-autoregressive multiresolution sequence imputation». In: *Advances in Neural Information Processing Systems*. 2019, pp. 11236–11246 (cit. on p. 14).

[56] Pengcheng Lin, Kejiang Ye, and Chang-Zhen Xu. «Using a multi-branch convolutional neural network for detecting unknown network attacks». In: *Future Generation Computer Systems* 109 (2020), pp. 418–435 (cit. on p. 14).

[57] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. «Connecting the dots: Multivariate time series forecasting with graph neural networks». In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 753–763 (cit. on p. 15).

[58] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. «Comparison of ARIMA and LSTM in forecasting time series». In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2018), pp. 1394–1401 (cit. on p. 15).

[59] Charles Spearman. «The proof and measurement of association between two things.» In: (1961) (cit. on p. 25).

[60] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: `1412.6980 [cs.LG]` (cit. on p. 28).