Cloud-Optimized Fusion of Time Series and Machine Learning Models for Enhanced Real-Time Forecasting

MSc Research Project
Cloud Computing

Sruthi Praveen
Student ID: x21223785

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sruthi Praveen
Student ID:	x21223785
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Aqeel Kazmi
Submission Due Date:	18/09/2023
Project Title:	Cloud-Optimized Fusion of Time Series and Machine Learning Models for Enhanced Real-Time Forecasting
Word Count:	6014
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sruthi Praveen
Date:	18th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Cloud-Optimized Fusion of Time Series and Machine Learning Models for Enhanced Real-Time Forecasting

Sruthi Praveen
x21223785

Abstract

This study assesses the effectiveness of Time-series based Machine Learning models in forecasting product sales using stock prices of major technology companies such as Google, Amazon, Microsoft, and Apple, as well as standard datasets including air passengers' data, shampoo sales, Big Mart sales, and a real-time dataset comprising 100 Stock Keeping Units (SKU's). Models tested include Linear Regression, Support Vector Machine, Decision Tree, Gradient Boosting Regression, and ARIMA. The system is designed for automated data processing and preparation to ensure data integrity. Mean Squared Error, Mean Absolute Error, and Mean Absolute Percentage Error are employed for model accuracy assessment. The findings guide businesses in selecting the optimal model for predicting product commercial future, contributing to informed decisions. The research introduces a novel approach merging clustering with model selection, enhancing time series forecasting precision. Cloud platform deployment enhances accessibility and usability. This approach not only improves accuracy but also yields interpretable forecasts, benefiting various domains seeking accurate and timely predictions.

1 Introduction

Cloud computing involves the delivery of computing services over the internet. A key component in cloud computing is the cloud instance, which is a virtual server resource hosted by an external cloud service provider. Using cloud instances eliminates the need for organizations to manage physical server infrastructure in-house. Cloud service providers maintain the hardware in their data centers and provide virtual instances that tap into the cloud's processing capabilities. Some key benefits of cloud instances include:

Scalability: The processing power of a cloud instance can be scaled up or down as needed by the user. For example, if an application sees increased usage, programmers can provision more resources for the instance hosting it.

Redundancy: Multiple copies of an instance can be made for backup purposes, providing redundancy within an organization. This is especially useful for memory-intensive tasks like data processing. For example, if an application is hosted in the cloud and one of its European instances goes down, it can keep running on instances in the US and Asia.

Time series forecasting is an important technique used by cloud service providers to optimize resource allocation and capacity planning. By generating accurate demand forecasts, cloud providers can anticipate surges in customer needs, allocate resources most efficiently, and avoid unnecessary overprovisioning. Examining historical performance data and deriving insights enables proactive mitigation of potential service degradation issues before they arise. The use of time series forecasting is thus a critical tool for enhancing the performance and availability of cloud services.

As time series forecasting is leveraged across diverse sectors, the techniques discussed in this research have versatile applicability. Accurate projections empower organizations across many fields to make informed decisions and strategic plans. This research explores approaches to enhance time series forecasting, which would equip cloud providers to better adapt to fluctuations in user demand and deliver consistently high-quality services.

1.1 Aim:

Forecasting consumer demand for emerging technologies is growing increasingly vital, as it enables leading tech companies to evaluate their potential for offering new services and guides investment priorities. Accurate demand predictions empower firms to determine the value of transitioning physical infrastructure to cloud-based platforms, which can reduce costs.

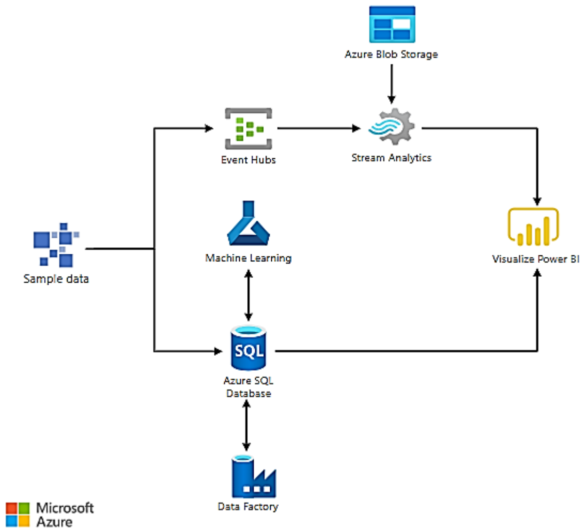

Figure 1: Cloud Infrastructure for the time series forecasting used in Microsoft Azure ***Demand Forecasting*** (n.d.)

1.2 Motivation:

Cloud service providers need to forecast how their customers will come and go so they can use their resources effectively, reduce their expenses, and maximise their productivity. The algorithms here make forecasts that help businesses expand their performances and infrastructure to attain peak efficiency and performance. To keep up with the needs of its numerous consumers, cloud services must guarantee and sustain a high level of customer satisfaction. There are limitations to time series models in terms of portability to the cloud, speed of execution, and the accuracy with which they replicate patterns.

They can be time-consuming and costly to train, as well as erroneous because of shifts in the underlying data patterns. They could also have trouble duplicating specific patterns, which could dampen the performance of the model. While these time series based Machine Learning models are helpful in forecasting, they are not without their limits. The speed may be greatly improved if a mechanism is developed to cluster the time series data based on the attributes of the data and then assigning the best model for that cluster. As a result, calculation time will increase, and fewer models will be executed.

1.3 Research Objective:

Given the current limitations of the models in terms of deploying them in the cloud , time of execution, and real-world pattern replication, improving these areas might be a worthwhile area of study. Given that these models require more time to run, this may be the intended outcome. New approaches of operating these models in cloud systems and improving the accuracy and reproducibility of patterns, might be part of the solution. Another possible objective of the study is to find the best practises for deploying and utilising these models in the cloud. Doing so would guarantee the best possible efficiency and performance. The goal of this research is to enhance the effectiveness of time series models in the deployed environment. As a result, cloud service providers would be better equipped to predict and adapt to consumer demand fluctuations and provide consistently high-quality service.

1.4 Research Questions:

The four existing potential research questions that integrate time series clustering concepts with the above research questions are as follows:

RQ1: In what ways can clustering enhance the precision and effectiveness of time series forecasting models within cloud environments, and what are the pivotal elements that contribute to the bolstering of this strategy?

Answer: In this research the model and the algorithms are trying to club different algorithms to find similar patterns in the data. In this each data in the club will be quite similar and this helps in forecasting the pattern correctly.

RQ2: What strategies can be employed to effectively combine time series clustering principles with various machine learning and artificial intelligence methodologies, aiming to optimize the effectiveness of cloud-based time series models?

Answer: The cloud-based models that are used here directly reflect the cloud consumption of the company. Each of major companies whether it is Meta, Google, Amazon or Microsoft are dealing with the cloud consumption. The whole solution designed over here are designed for the diversified datasets.

RQ3: How might the utilization of time series clustering outcomes, enhance the ability of models to recognize patterns within real-time, unseen actual data?

Answer: The research found that each algorithm tends to predict one type of pattern well. When time series data is clustered into groups containing similar patterns, the algorithm assigned to that cluster can generate more accurate forecasts for unseen, real-time data. The analysis showed that clustering time series based on pattern attributes enables models to recognize those patterns in new data more effectively. Grouping data with comparable characteristics allows the matched algorithm to produce improved predictions for that cluster's future values.

RQ4: How might the utilization of time series clustering concepts enhance the comprehensibility of time series models in cloud environments, and what implications does this hold for both cloud service providers and end-users?

Answer: For major cloud providers like Microsoft, Amazon and Google, a significant revenue component comes from consumption of their cloud services. Cloud usage directly impacts company financials and stock valuation. By clustering time series data based on key attributes like trend, seasonality and level, models can gain better insights into customer demand patterns. The analysis here clustered 1209 stock keeping units (SKUs) in under a minute using API deployment on cloud infrastructure. This leverages autoscaling, load balancing and multithreading for large-scale clustering, making trends across thousands of time series more visible through stable analysis.

For cloud providers, enhanced model comprehensibility supports better visibility into usage patterns and demand drivers. This aids decision making around areas like resource allocation and capacity expansions. Increased transparency also facilitates early identification of anomalous clusters, enabling model re-tuning.

On the user side, interpretable forecasting models provide confidence in projection reliability and accountability. This gains customer trust and satisfaction. When models can better replicate real-world data patterns through clustering, their predictions become more credible and useful in planning.

This research conducted a literature review of 19 papers related to time-series and cloud forecasting. Section 3 proposes the methodology on which this study is based. Section 4 examines the design specification. Section 5 discusses the implementation through comparative analysis of the models used, highlighting the most efficient one. Section 6 analyzes the results obtained from this research and discusses the ethical considerations. Finally, Section 7 highlights the conclusions of this study.

2 Related Work

Due to their ability to grasp intricate patterns and trends within data, sophisticated models rooted in deep learning, such as artificial neural networks, deep neural networks, and recurrent neural networks, exhibit remarkable performance in predicting time series. Among these, Long Short-Term Memory (LSTM) models, a variant of recurrent neural networks, have demonstrated exceptional proficiency in addressing time series prediction tasks. These models selectively decide what information to retain or discard from prior time steps. The incorporation of deep learning models contributes to the enhancement of precision and dependability in time series forecasting models, particularly in scenarios where conventional statistical methodologies might prove inadequate (Islam, S.M. et al., 2023). (**Islam et al. (2022)**)

2.1 Recent trends on Artificial Intelligence and machine Learning towards time series forecasting

A cutting-edge AI system has been developed to accurately predict future power consumption. Prior to inputting data into the prediction layers of the multilayer perceptron, the system is divided into three components: effective data preprocessing, convolutional long short-term memory (ConvLSTM), and a bidirectional gated recurrent unit (BD-GRU). The performance of the newly suggested model surpasses that of current leading methods, showcasing a mean square error (MSE) in the range of 0.012 to 0.045 for hourly data. Empirical testing using actual energy data from residential and PV systems demonstrated the superiority of the proposed model when compared to existing techniques. (**Khan et al. (2023)**

The research conducted by **Masini et al. (2023)** explores advanced approaches for time-series prediction and supervised machine learning. The study delves into techniques like penalized regressions, model ensembles, tree-based methods, as well as feedforward and recurrent neural networks. Additionally, the research covers ensemble and hybrid models. The paper investigates various metrics that signify strong predictive capabilities, while also demonstrating the application of machine learning in the fields of economics and finance. To illustrate these concepts, the authors employ a high-frequency financial dataset as a practical case study.

The approach introduced by the authors **Jiang et al. (2023)** offers a novel perspective on understanding shifts in the pace of the epidemic's advancement, including identifying points of transition. Their method employs a self-normalization (SN) strategy to simultaneously assess these transitions. The application of this technique to a selection of 30 prominent nations unveils intriguing trends that could potentially influence strategies for addressing pandemics. Furthermore, the paper presents a dual-phase prediction method that utilizes a change-point identification algorithm in tandem with an extrapolation function to estimate overall mortality in the context of the United States.

The innovative solution proposed involves synergizing the Prophet system, developed by Facebook (**Jun-Gang (2023)**), with the LSTM model from long-short memory networks. While the Prophet system is potent for time series forecasting, its precision and capability to capture intricate time series attributes have constraints. This inventive approach is rooted in enhancing the Prophet model, leveraging its strengths, and augmenting them through integration with the LSTM model. This merging, results in several benefits, such as improved forecast precision and the ability to uncover intricate patterns within time series information.To illustrate the effectiveness of this combined technique, it is applied to predict temperature patterns using the Shanghai temperature dataset. The outcomes underscore the prowess of the hybrid model as an exceptional tool for forecasting, successfully addressing the aforementioned limitations.

In the recent work by **Raykar et al. (2023)**, a novel method named TsSHAP is presented, which aims to enhance the interpretability of time series forecasting models. This technique offers a way to explicate the predictions of any black box forecasting model by utilizing user-defined interpretable factors. Importantly, this approach remains unaffected by the specific forecasting model employed, delivering explanations based on SHAP values derived from the application of the TreeSHAP algorithm to a surrogate model. This work contributes to the standardization of local, semi-local, and global explanatory concepts within the realm of time series forecasting. Through extensive experimentation across diverse datasets, the effectiveness and robustness of TsSHAP

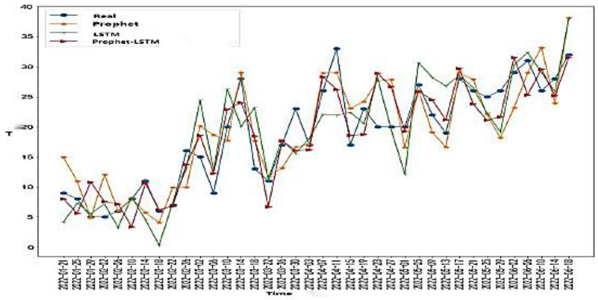

Figure 2: Comparison of the different models used in (**Jun-Gang (2023)**)

have been substantiated.

The research conducted by **He et al. (2023)** introduces an innovative strategy to predict financial time series using a deep learning ensemble approach. This technique integrates a combination of linear and nonlinear data characteristics commonly found in financial time series. The ensemble model is comprised of a convolutional neural network (CNN), a long short-term memory network (LSTM network), and an autoregressive moving average (ARMA) model. The CNN-LSTM component captures the spatial and temporal attributes of the data, while the ARMA model is employed to account for autocorrelation patterns. Empirical observations demonstrate that the proposed approach outperforms individual benchmark models in terms of both prediction accuracy and robustness. This study underscores the significance of financial time series forecasting given the continuous expansion of global financial markets in response to rapid occurrences like climate shifts and worldwide warming. The research highlights the critical role of accurate financial time series prediction in facilitating the effective operation and management of financial markets.

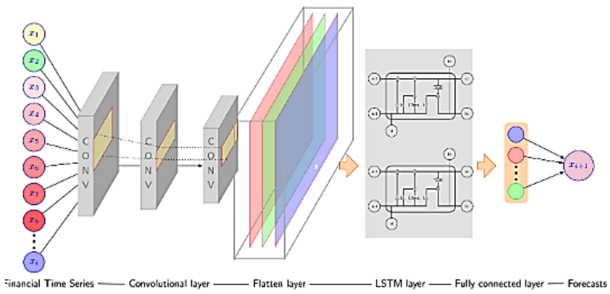

Figure 3: CNN-LSTM Network used in (**He et al. (2023)**)

The research conducted by **Semenoglou et al. (2023)** explores the utilization of data augmentation methods to enhance the effectiveness of neural networks in predicting univariate time series under conditions of limited data availability. The study leverages datasets from the M4 and Tourism competitions and evaluates nine distinct strategies for enhancing data, encompassing adjustments, generative models, and up-sampling techniques. The outcomes propose a notable enhancement in prediction accuracy by synergizing deep neural networks with data augmentation strategies, particularly those rooted in up-sampling and amalgamation of time series data. This synergy proves especially valuable when dealing with constrained initial sizes of training datasets.

In their recent study (**Lazcano et al. (2023)**) introduced an innovative strategy for enhancing time series forecasting. The approach involves the integration of two distinct neural network architectures: a Graph Convolutional Network (GCN) and a Bidirectional Long Short-Term Memory (BiLSTM) network. This amalgamation exhibits notable advancements over earlier outcomes. The research assesses the combined model's efficacy against both separate BiLSTM and GCN models, along with traditional forecasting models. Various evaluation metrics, including RMSE, MSE, MAPE, and the R-squared value, were employed to gauge performance. The outcomes of this investigation highlight that the proposed composite model yields more precise predictions while concurrently reducing prediction errors.

The research conducted by **Adhikari et al. (2019)** proposes an ensemble model combining time series and machine learning techniques to improve demand forecasting accuracy. The model handles data preprocessing, outlier detection, missing values, and ensembling forecasts from ARIMA, regression algorithms like SVR and random forest. It is evaluated on sales data from multiple markets, showing improved and more stable results than traditional statistical methods.

2.2 Summary of different Machine Learning and A.I trends in forecasting

The research examined encompasses a range of AI frameworks that blend different approaches, intricate models designed for high-dimensional data, algorithms aimed at detecting changes in patterns, methods that explain predictions based on features, and strategies to enhance datasets. These investigations thoroughly evaluate the effectiveness of various neural network architectures—such as ConvLSTM, BDGRU, LSTM, and GCN—when compared to established models. The results of these analyses reveal that these proposed models enhance both prediction accuracy and the ability to handle unexpected situations, surpassing established methods in real-world scenarios. Additionally, these studies underscore the significance of time series prediction across diverse fields, encompassing energy, economics, finance, and healthcare. They stress the critical need for precise and dependable forecasts in these areas.

2.3 Impact of Segmentation towards time series forecasting

The study conducted by **Lafabregue et al. (2022)** explores the application of time-series aggregation to enhance the performance of energy systems, particularly in the context of optimization models. The authors offer practical suggestions for effectively utilizing time-series aggregation, including thorough performance monitoring, customizing techniques to address specific challenges and data characteristics, and exploring various clustering strategies. The research also delves into the utilization of deep clustering to organize time series data, deconstructing this process into components like network design, pretext loss, and clustering loss. Through an examination of 300 diverse models from the UCR/UEA time series dataset repository, the researchers compare these models against existing clustering methods. Ultimately, they introduce an approach for identifying patterns within network classifications of time series data using class activation mapping, all within an unsupervised context.

The study by **Zhou et al. (2022)** presents an innovative approach aimed at restoring lost information within time-series imagery from Landsat satellites, a challenge often caused by cloud cover or shadows. This novel technique capitalizes on deep learning, harnessing a combination of autoencoding, long-short-term memory (LSTM), and a specialized backward LSTM-based approach for time-series prediction. The methodology involves the creation of pixel-level time-series data through manual application of cloud and shadow masks onto multi-temporal satellite images. The restoration of missing data points in these time-series is achieved through an unsupervised AE-LSTM strategy, with the added development of a dedicated bidirectional LSTM model for each cluster. The effectiveness of this approach was demonstrated across three distinct Landsat-8 OLI datasets. Encouragingly, the results exhibited an enhancement of over 10% in normalized mean-square error in comparison to prevailing state-of-the-art methodologies.

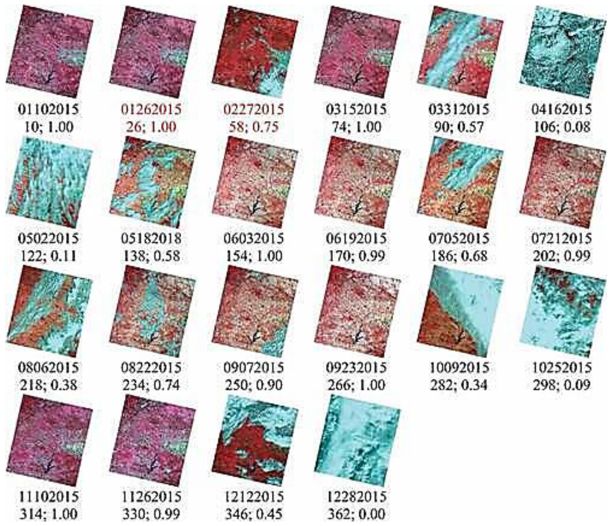

Figure 4: Image based time series data and the way they are clustered (**Zhou et al. (2022)**)

The study conducted by **Levantesi et al. (2022)** introduces a novel approach to studying life expectancy by integrating functional clustering with Long-Short Term Memory (LSTM) neural networks. Utilizing non-parametric smoothing, they generate life expectancy curves for diverse nations, cluster these curves, and employ a multivariate LSTM neural network to simultaneously forecast life expectancy within each cluster. The findings indicate consistent life expectancy patterns in industrialized nations, while the cluster representing medium-low longevity exhibits notable cross-country variations. Notably, the LSTM method demonstrates superior predictive accuracy compared to the standard VAR model. This combined employment of functional clustering and neural networks enhances the comprehension of life expectancy trends.

2.4 Summary

The research conducted by **Lafabregue et al. (2022)** explores the application of time-series aggregation and deep clustering for the organization of time series data. The study emphasizes comprehensive performance evaluation, customization of approaches, and assessment of clustering methodologies. By utilizing the UCR/UEA repository of time series datasets, the authors meticulously assess 300 distinct models, comparing them with existing clustering methods. Additionally, they introduce an innovative method for utilizing class activation mapping in an unsupervised context, enabling the identification of patterns within network classifications of time series data. In a separate work by **Zhou et al. (2022)** deep learning and a clustering strategy are employed to restore missing data in time-series satellite images, resulting in superior outcomes compared to state-of-the-art techniques. **Levantesi et al. (2022)** integrate functional clustering and LSTM neural networks to predict life expectancy. Their findings reveal that the Long-Short Term Memory approach surpasses the conventional VAR model, offering deeper insights into the evolution of life expectancy trends. Collectively, these studies underscore the effectiveness of clustering techniques in enhancing the precision and performance of time-series analysis.

2.5 Cloud Platforms implementation towards time series forecasting

In the publication authored by **Rigakis et al. (2023)**, a novel approach is introduced for evaluating life expectancy by integrating functional clustering with Long-Short Term Memory (LSTM) neural networks. Through the utilization of non-parametric smoothing, life expectancy curves are generated for diverse nations and subsequently grouped, offering a distinctive viewpoint on the trajectory of life expectancy changes. Notably, the analysis uncovers a consistent pattern in life expectancy among industrialized nations, while also highlighting substantial heterogeneity across countries in the medium-low longevity cluster. Remarkably, the LSTM technique demonstrates superior predictive accuracy compared to the conventional VAR model.

The research conducted by **Teichgraeber and Brandt (2022)** delves into the realm of time series forecasting within cloud-based environments. The study highlights the benefits of leveraging cloud platforms, including scalability, cost-effectiveness, and user-friendly features. Leading platforms such as AWS, Azure, and GCP offer a range of tools for various time series forecasting tasks, spanning data storage, preprocessing, model training, and deployment. The research also advocates for cloud-based frameworks like Amazon SageMaker, Azure Machine Learning, and Google Cloud AI Platform, which provide both ready-made algorithms and customizable options for time series forecasting. The authors emphasize the importance of aligning chosen tools with specific forecasting requirements. Real-world examples from diverse industries are utilized to illustrate these principles. Ultimately, the study underscores the substantial advantages that cloud-based platforms and frameworks bring to time series forecasting, enabling organizations to efficiently develop and implement scalable, cost-effective, and accurate forecasting models.

2.6 Summary

The preceding studies highlights the advantages of employing cloud-based platforms for deployment of time series forecasting projects, such as scalability, cost-effectiveness, and ease of use. According to researchers, cloud-based platforms like as AWS, Azure, and GCP offer a variety of features that may be used for time series forecasting activities such as data storage, pre-processing, model training, and deployment. They also recommend cloud-based frameworks like Amazon Sage Maker, Azure Machine Learning, and Google

Cloud AI Platform, which offer pre-built algorithms and tools for time series forecasting as well as the option to customise and develop your own models. The authors stress the necessity of selecting appropriate tools and services based on the unique needs of the time series forecasting activity, and they present examples of real-world use cases from diverse sectors.

2.7 Research Niche

The research publications cited earlier reveal a scarcity of efforts directed towards the clustering of time series data based on data patterns, followed by the allocation of more suitable procedures to these clusters. Time series data clustering represents a strategy to enhance forecast accuracy, encompassing the grouping of similar time series and subsequent application of relevant algorithms to each cluster based on their unique attributes. This approach holds the potential to discern distinct patterns and trends within time series data, leading to more refined projections. The implementation of this methodology onto a cloud platform could involve the utilization of technologies such as Apache Spark, Hadoop, and Kubernetes. These technologies offer the capability to manage large datasets and harness the potential of machine learning algorithms to enhance forecasting models.

3 Methodology

3.1 Research Method

This section discusses in detail the methods applied for forecasting the data. Below is an overview of the pipeline of the model by which it functions. Fig.5

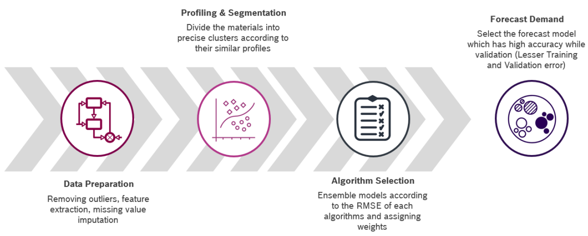

Figure 5: An overview of the model pipeline

3.1.1 Data Acquisition

The metrics being forecasted in this research include revenue, hours, count, and more. Since a single transformation cannot accommodate all data types effectively, this study employs various variable transformations to attain stationarity and stable projections across the diverse metrics. Some metrics contain a mix of positive and negative values, further complicating analysis. To address this challenge, the transformations are tailored to each metric to enable unified forecasting. The datasets utilized in this research analysis include:

1. Daily Time Series Dataset: `https://finance.yahoo.com/quote/%5EGSPC/history`

a. MSTF – Microsoft

b. GOOG – Google
c. AMZN – Amazon

2. Shampoo Sales - `https://github.com/jbrownlee/Datasets/tree/master`

3. Air Passenger Dataset - `https://github.com/jbrownlee/Datasets/blob/master/airline-passengers.csv`

3.1.2 Data Pre-processing

a. Data determination - The data that arrives could have been monitored over various time spans, such as hourly, daily, weekly, monthly, yearly etc. The time intervals and the nature of the time series have been documented for this data.

b. Data Imputation – Before utilizing the data for model training, the initial step involves addressing missing values through data imputation.

Missing Value Imputation – This process takes into account both the dataset's size and the frequency of missing values. When the count of missing values surpasses a specific threshold, these gaps are filled with zeros. Conversely, when only a small number of values are absent, a tailored method is applied for imputation. This approach involves leveraging neighboring data points and corresponding data from adjacent time periods – either past or future – provided they are accessible. Additionally, for scenarios involving the introduction of new products, data is sourced from the initial non-missing entry.

a. Missing Dates Imputation – The data is categorized based on its interval type, which might include options like daily, weekly, monthly, or yearly. Any gaps in the data timeline are filled in by imputation. The sales data also undergoes imputation to handle missing values.

b. Outliers Treatment – This study implements Outlier handling on dataset segments selected based on the data's interval type or windowing. Within each window, outliers are identified using the Median Absolute Deviation (MAD) or Mean Standard Deviation (MSD) methods. Subsequently, each identified outlier is closely monitored to determine whether it exhibits a seasonal pattern in the upcoming corresponding timeframes, spanning various years. In instances where such irregularities or outliers showcase this seasonal behavior, they are constrained by setting upper and lower boundaries derived from MAD and MSD computations. Emphasis is placed on prioritizing MAD, and if its application proves unfeasible, then MSD is employed as an alternative approach. As an illustration, if the data is monthly, the outlier treatment is enacted over a span of 12 months. The data undergoes Median Absolute Deviation or Mean Standard Deviation analysis, and data points deviating by more than 3 standard deviations from the mean are trimmed.

The equations are:

$$MSD = \alpha \cdot \mu \pm \beta \cdot \delta \tag{1}$$

Where α and β are constants and μ is the mean, and δ is the standard deviation.

$$MAD = \alpha \cdot Median \pm \beta \cdot MedianAbsoluteDeviation \tag{2}$$

c. Determination of Lags – The determination of lags involves analyzing ACF (Auto Correlation Plots). These plots unveil the connection between two previous-time observations, revealing both direct and indirect dependencies. PAF (Partial Auto Correlation) plots, on the other hand, reveal the partial correlation between a stationary time series and its lagged values. This insight exposes the direct link between an observation and its lag. Consequently, the ACF and PACF plots play a crucial role in identifying the ideal values for the parameters p, d, and q. The count of lags is represented as p, while the order of the moving average model is denoted as q. If the ACF and PACF plots exhibit a gradual decline, the time series necessitates transformation into a stationary state, facilitated by the utilization of d, signifying the degree of differencing.

d. Stationary Data – Three basic criteria classify data as stationary:

• The meaning of the series should not be a function of time. It should be constant.

• The variance of the series should not be a function of time. This is called homoscedasticity.

• The covariance of the ith term and the $(i + m)$th term should not be a function of time (t).

The data is made stationary by differentiating the dataset using a lag. The stationarity of the dataset is verified using the Dickey-Fuller Test (which can be set as the value of 'd' in the case of the Hyperparameter of ARIMA).

e. Supervised Learning Dataset – The stationary data is overseen through the utilization of lags, which are determined using Auto Correlation Plots, and these lags are represented by the hyperparameter "p" in the ARIMA model.

Flask API: The Flask API is used to implement the time series forecasting web service. It handles the overall workflow of accepting data uploads, generating forecasts, and returning predictions via a JSON API.

4 Design Specification

4.1 Research Resources – Machine Learning Algorithm

The prepared dataset is subjected to machine learning and time series techniques. This group of algorithms is used to choose the best models.Ensemble technique is applied to this group of algorithms in order to characterise the bias or variation in the approach and improve forecasting accuracy.

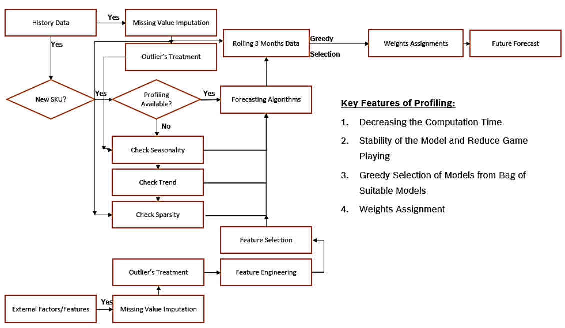

Figure 6: Model Flow chart

Most Time Series and Machine Learning approaches are considered here, so that one piece of data that may not function well in one of the algorithms can be used in the other ways. Also, an ensemble technique averages the forecasts from the times series forecasting technique and machine learning forecasting techniques and modifies the bias and variance thus bringing the forecast close to the actual sales used. The preceding findings show that the ensemble of the results of the time-series model and regression-based model produces a superior result owing to the fact that it eliminates over-forecasting and under-forecasting and brings the forecast values closer to the actual values. These findings outperform the individual algorithms utilised in the two models. These results are also consistent and dependable for long-term projections.

Step 1: Cross Validation Sets - The training and validation sets are divided in a 7:3 or 8:2 ratio. RMSE (root mean square deviation) is used to verify the error term. Following several iterations and study, RMSE was determined as a better method for calculating errors. Based on the windowing approach, the validation set is further segmented into three sets.Example, Training dataset: ID's 0:30 and Validation Dataset: ID's 31: 35.

Hence for each Model,
T1: 0: 30, V1: 31: 33
T2: 0: 31, V2: 32: 34
T3: 0: 32, V2: 33: 35

The performance of each model is selected based on the volatility factor which is calculated from the mean and standard deviation from all three validation RMSE values.

Step 2: Feature Selection - Random Forest Regression is employed utilizing its default configurations or fine-tuned hyperparameters, to carry out the process of feature selection.

Step 3: Data Profiling – This phase holds significant importance within the context of data clustering. When dealing with Time Series Data, it's crucial to recognize its three fundamental components:

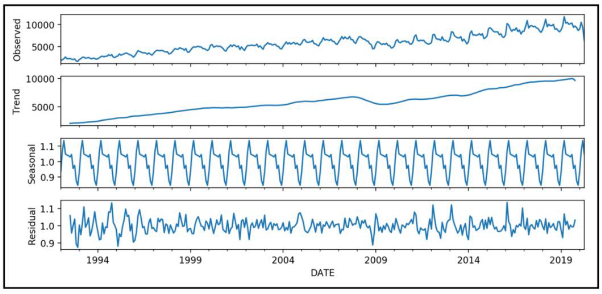

Figure 7: Components of Time Series Dataset
Time Series Decomposition (n.d.)

a. **Trend:** A time series dataset consists of data points recorded over time, where each point corresponds to a specific time interval. Within such datasets, there exists a prolonged pattern or trajectory that persists over time. This pattern is referred to as the "trend." Trends within time series data come in three variations: upward, downward, and stationary.

• **Upward trend:** An upward trend materializes when the data points within the time series dataset exhibit a consistent increase over time. This signifies that the variable in focus is gradually rising over time.

• **Downward trend:** Conversely, a downward trend takes shape when the data points in the time series dataset showcase a consistent decrease over time. This suggests that the variable being observed is on a declining trajectory.

• **Stationary trend:** The concept of a stationary trend comes into play when the data points display neither significant growth nor decline over time. This indicates that the attribute under scrutiny remains relatively constant or stable across the time period.

b. **Seasonality:** Recognizing seasonality within a time series dataset involves observing its evolution over time to identify consistent patterns of variation. These patterns typically manifest as recurring cycles of peaks and valleys, manifesting at regular time intervals. These intervals, known as seasonal cycles or frequencies, unveil the rhythm of these patterns. Once identified, these seasonal patterns can be harnessed to predict forthcoming data points within the time series. The presence of seasonality in such datasets is of great significance, offering insights into the inherent trends and behaviors of the studied variable. Moreover, this knowledge aids in forecasting future values, facilitating informed decision-making and strategic planning. Additionally, by detaching the seasonal element from the time series data, the potential emerges to uncover supplementary trends or patterns, classifiable into three categories:

• Complete Seasonal
• Partial Seasonal
• No Seasonal

c. **Noise or Residual:** This represents an undesirable element within the dataset.

Apart from this, there's an additional factor known as "Level" that signifies the dataset's specific position, crucial in time series forecasting. By employing these criteria, the dataset can be categorized into three distinct trends, three types of seasonality, and the particular data point value utilized in predictions. The cluster can look like this:

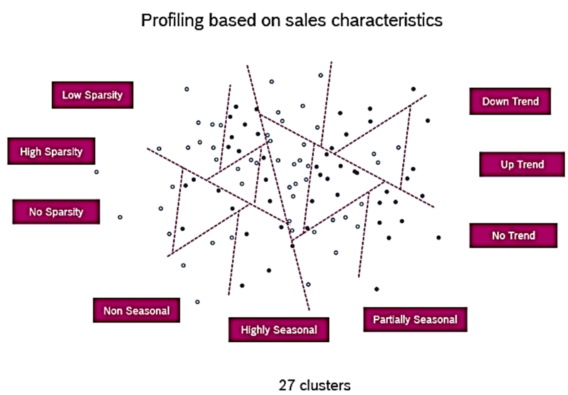

Figure 8: Clustering Example

Step 4: Sparse Data Handling - Intermitted Demand occurs when a product suffers erratic demand and many periods of zero demand. It's common in areas including aviation, automotive, defence, and manufacturing. It also happens with items towards the end of their life cycle. In these instances, the forecasting difficulty is the uncertainty in predicting when the next demand will come and the level of demand when it does. The Croston model is used for items that have intermittent demand.

Step 5: Time Series Algorithms - The datasets are subjected to ARIMA (Auto Regressive Integrated Moving Average), ARMA (Autoregressive Moving Average), AR, Moving Average, Weighted Moving Average, Exponential Time Smoothing, Holt-Winters, and other models. If the outcome is negative, the weighted moving average is used. The best model from the validation set is chosen and saved alongside the mistake.

Step 6: Machine Learning Algorithms - These algorithms use differenced and scaled data. The data is different when d is used, as estimated by ACF and PACF plots. On the datasets, several machine learning techniques such as Linear Regression, Decision Tree Regression, and SVM are used. To tweak the hyperparameters, grid search and random search are used. The scaling strategy is used depending on the algorithm.

Step 7: Ensemble Algorithms - The weight computations are based on the errors derived from the most effective models within the Time Series ensemble of models and the Machine Learning collection of regression-based algorithms. When utilizing deep learning models, they are categorized as models falling within the scope of deep learning. The weights are computed as follows:

$$W_{ts} = \frac{1}{error_t}$$

$$W_{ts} = \frac{\frac{1}{error_t}}{\frac{1}{error_t} + \frac{1}{error_m} + \frac{1}{error_{dl}}}$$

$$W_{ml} = \frac{\frac{1}{error_{ml}}}{\frac{1}{error_t} + \frac{1}{error_m} + \frac{1}{error_{dl}}}$$

$$W_{dl} = \frac{\frac{1}{error_{dl}}}{\frac{1}{error_t} + \frac{1}{error_m} + \frac{1}{error_{dl}}}$$

These weights are multiplied with respective forecasts developed by the forecasting engines and the final forecasts are calculated.

5 Implementation

This research created a comprehensive framework in Python to tackle the forecasting challenges. The framework employs various Python libraries to effectively fulfill the objectives. Historical sales data was collected, forming the basis for the forecasting models. Libraries like pandas, numpy and datetime were used for data manipulation, numerical computations and handling time information respectively.

Multiple approaches were implemented to develop accurate sales forecasting algorithms, each utilizing specific machine learning techniques. These included time series analysis leveraging techniques like Moving Averages, Exponential Smoothing and ARIMA to capture trends and seasonality. Machine learning regressors like Random Forest, Support Vector Regression and Gradient Boosting Regression were utilized to model intricate patterns in real-world time series data. Their ensemble and non-linear fitting capabilities enabled capturing complex relationships. Random Forest and Gradient Boosting generate robust forecasts by reducing variance through aggregating multiple base learners. They also provide greater model interpretability, offering transparency into important features and relationships, aligning with the research objective.

Synthetic datasets were generated based on historical sales patterns to evaluate the forecasting algorithms. These combined trends, seasonality and random fluctuations to simulate real-world dynamics. The threading library enabled concurrent processing and real-time capabilities. The matplotlib library provided graphical visualization for analysis.

The core research aim is to improve business planning through accurate sales predictions, by minimizing over and under estimations. The framework assists inventory management, resource allocation and strategic decisions. The architecture mirrors real-world sales dynamics to ensure practical applicability. Overall, each component serves a specific purpose, contributing to the success of sales forecasting.

5.1 Research Resource – Cloud Implementation

In this whole research the models are implemented in Cloud using AWS.

5.1.1 AWS

There are various processes involved in setting up an AWS system for forecast creation. The general steps include:

- **Sign up for an AWS account** if one does not already exist.
- **Choose an appropriate forecasting tool** from the options AWS offers like Amazon Forecast, AWS Glue and Amazon SageMaker. Select the tool that best fits the needs.
- **Prepare the data** by cleaning, transforming and formatting it in a way suitable for forecasting. This includes outlier removal, filling data gaps, standardizing data etc.
- **Create a forecasting model** using the chosen tool. This involves carefully selecting the method, tuning hyperparameters and training on historical data.
- **Evaluate the model** by comparing its predictions to real data. This helps detect discrepancies and fine-tune the model.
- Once model performance is satisfactory, deploy it to start generating live forecasts. This may require infrastructure like AWS Lambda functions or Step Functions to streamline the workflow.

The exact steps will vary based on the specific forecasting tool and dataset. AWS tools were not used in this research since the focus was on understanding the underlying techniques, methodologies, and their implications, rather than solely relying on a pre-built tool. It is important to closely follow AWS guidelines and best practices to ensure the system is properly configured and generates accurate forecasts.

5.2 Implementation of Flask API:

The API is structured around a '/forecast' endpoint that responds to POST requests. When called, it executes the below steps:

1. The uploaded CSV file is retrieved from the request and saved locally. Additional parameters like the forecast model type and prediction period are extracted from the POST data.
2. The pandas library is leveraged to load the CSV into a DataFrame. This allows analyzing the structure and manipulating the data.
3. A loop iterates through each unique key or identifier in the DataFrame, assuming these distinguish separate time series.
4. Inside the loop, the subset of data for that key is extracted and wrangled into a format suitable for forecasting. This involves transposing it into a single column.
5. The preprocessed data feeds into a reusable predict function to generate forecasts

based on the requested model type and time period.
6. All predictions are consolidated into a final DataFrame where each column maps to a time series key.
7. The forecast DataFrame is encoded into JSON for returning in the API response.
8. Flask configures the web application with the defined route, host, and debug mode.

Fig.9 illustrates the API run in Postman.

Link for Flask API:
`https://api.postman.com/collections/12533933-85644ef0-ac1a-4fb8-9986-e8256dc59d4b?access_key=PMAT-01H7H57YC0CRQWFE1FTA9T17MT`

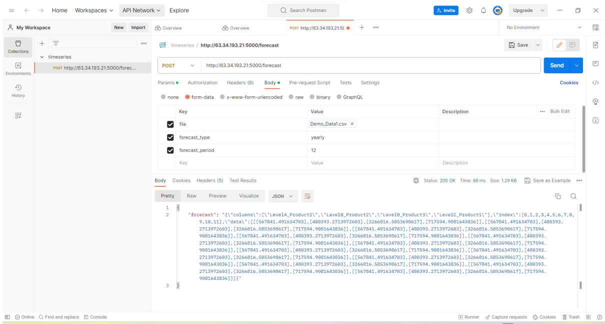

Figure 9: Implemented Flask API

6 Evaluation

6.1 Evaluation Procedure

For precise measurements, each sector has its own metric. FACC (Forecast Accuracy) and MAPE (Mean Absolute Percentage Error) are two such measurements.

$$FACC = 1 - \sum_{k=1}^{n} \frac{|actuals - forecasts|}{actuals}$$

$$MAPE = \frac{(\sum |actual - forecast|/actual) \times 100}{forecast\, period}$$

6.2 Evaluation of the metrics

The Figure.10 presents the performance evaluation of the machine learning (ML) model using different metrics (SKU's) provided by the company Analytic Labs, comparing the forecasts generated by the model against the actual data. The evaluation is conducted for two quarters (FY22 Q2 and FY22 Q3) and involves two forecasting methodologies: "FaaS" (Function-as-a-Service) and "Production" (a traditional production-based approach). The SKU's are the individual areas of cloud consumption values. The Production is the forecast generated for Sep-Dec 2022 (Q2 FY22) and Jan-Mar 2023 (Q3 FY22) that was used with SAP-APO.

Metric Name: This column specifies the name of the metric being evaluated, such as "SK1", "SK2", "SK3" and so on.

FYQ2 Wtd MAPE: and FY22 Q3 Wtd MAPE: These columns show the Weighted Mean Absolute Percentage Error (Wtd MAPE) values for the metric in question during the second and third quarters of FY22 (financial year 2022) for both the FaaS and Production forecasting methodologies.

Values: Historical data of the metrics for FY22 Q2 and FY22 Q3

Forecasted Values: Forecasts generated by the ML model using FaaS and Production methodologies

Calculation of Absolute Percentage Error (APE):

For each forecasted value, the value of APE is calculated using the formula:

$$APE = \frac{|forecastedvalue - actualvalue|}{actualvalue} \times 100$$

Weighted Mean Absolute Percentage Error (Wtd MAPE) Calculation:

1. The APE values and the weights are assigned to each metric to calculate the Wtd MAPE for both methodologies (FaaS and Production).
2. The weights are assigned based on the significance or contribution of each metric.

$$WtdMAPE = \frac{\sum(APE * Weight)}{\sum(Weight)}$$

Metric Name	FY22 Q2 Wtd		FY22 Q3 Wtd	
	Production**	FaaS	Production**	FaaS
SK1	9%	3%	3%	2%
SK2	25%	11%	25%	4%
SK3	15%	4%	13%	3%
SK4	21%	4%	12%	3%
SK5	19%	3%	29%	1%
SK6	12%	8%	8%	3%
SK7	9%	6%	9%	2%
SK8	11%	5%	5%	1%

Figure 10: Results analysis between the proposed FaaS and the model currently deployed in the production (SAP APO)

6.3 Final Forecast

The final forecast selected by the user is saved in the portal and in the newly downloaded .csv for reference. It can accommodate various time granularities such as Daily, Weekly, Monthly, Yearly and can effectively handle different kind of patterns. It ensures that the forecasts generated do not result in linear projections. The product adapts its predictions to actual outcomes using the weighted equations. Then the Best-of-the-Best (BoB) model is selected using the Greedy Selection process.

Since cloud providers rely on cloud-based consumption data to anticipate future trends, the model developed in this research provides accurate forecasts, enhancing visibility into future cloud consumption patterns. This equips cloud providers to make informed decisions regarding resource allocation and planning.

6.4 Ethical Considerations of the research

Any information of the company will not be divulged in this study; instead, it will be disguised and utilised to generate forecasts.

7 Conclusion and Future Work

Forecasting gives useful and trustworthy information about probable future occurrences and their implications for the organisation. This knowledge might help the organisation take actions to prevent future issues and ambiguity. As a result, providing accurate projections is critical. According to the results, greedy selection of the best models from each bag and greedy selection forecast from the time series and machine learning forecasting techniques provide a better forecast as it modifies the bias and variance, bringing the forecast closer to the actual sales. The Python platform is used to create the models. Almost all the approaches or models of Time Series and Machine Learning are addressed here, so that if one piece of data that may not function well in one of the algorithms can be used in the other ways.

While this research focused on custom implementations for exploring methodologies, an interesting extension would be utilizing AWS tools like Amazon Forecast and comparative benchmarking. Additional future work could involve advanced clustering to group series with similar attributes and behaviors to further enhance pattern recognition. Ongoing model tuning and testing across real-world datasets can build on the forecasting framework proposed here.Some of the USPs are listed below:

- The Weights Assigned are a penalty factor for the worst-performing algorithms and a reward for the better-performing Algorithms
- With the Amount of Data Volume Increasing over time, the DL algorithms weights will increase, and this will try to capture the harmonics within the data
- The Feedback System in the 3 Rolling Forecast tries to check for a stable model over the validation sample data
- Measure to check the forecast of each of the top contributing product series to see the variation and the forecast for each month/period not only in the forecast period but in the Validation period as well.
- Greedy Search works not only to find the best of TS, ML-Reg or DL but rather between the Best-of-TS, Best-of-Ml-Reg, and Best-of-DL along with the Ensemble in the 3 Rolling Validation Forecast.

References

Adhikari, N. C. D., Domakonda, N., Chandan, C., Gupta, G., Garg, R., Teja, S., Das, L. and Misra, A. (2019). An intelligent approach to demand forecasting, *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018*, Springer, pp. 167–183.

Demand Forecasting (n.d.).
URL: *https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/demand-forecasting*

He, K., Yang, Q., Ji, L., Pan, J. and Zou, Y. (2023). Financial time series forecasting with the deep learning ensemble model, *Mathematics* **11**(4): 1054.

Islam, S. M., Thakur, N., Garg, K. and Gupta, A. (2022). A recent survey on lstm techniques for time-series data forecasting: Present state and future directions, *Applications of Artificial Intelligence, Big Data and Internet of Things in Sustainable Development* pp. 123–132.

Jiang, F., Zhao, Z. and Shao, X. (2023). Time series analysis of covid-19 infection curve: A change-point perspective, *Journal of econometrics* **232**(1): 1–17.

Jun-Gang, G. (2023). Time-series prediction research based on combined prophet-lstm models, *Technical report*, EasyChair.

Khan, S. U., Khan, N., Ullah, F. U. M., Kim, M. J., Lee, M. Y. and Baik, S. W. (2023). Towards intelligent building energy management: Ai-based framework for power consumption and generation forecasting, *Energy and Buildings* **279**: 112705.

Lafabregue, B., Weber, J., Gançarski, P. and Forestier, G. (2022). End-to-end deep representation learning for time series clustering: a comparative study, *Data Mining and Knowledge Discovery* **36**(1): 29–81.

Lazcano, A., Herrera, P. J. and Monge, M. (2023). A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting, *Mathematics* **11**(1): 224.

Levantesi, S., Nigri, A. and Piscopo, G. (2022). Clustering-based simultaneous forecasting of life expectancy time series through long-short term memory neural networks, *International Journal of Approximate Reasoning* **140**: 282–297.

Masini, R. P., Medeiros, M. C. and Mendes, E. F. (2023). Machine learning advances for time series forecasting, *Journal of economic surveys* **37**(1): 76–111.

Raykar, V. C., Jati, A., Mukherjee, S., Aggarwal, N., Sarpatwar, K., Ganapavarapu, G. and Vaculin, R. (2023). Tsshap: Robust model agnostic feature-based explainability for time series forecasting, *arXiv preprint arXiv:2303.12316* .

Rigakis, I., Potamitis, I., Tatlas, N.-A., Psirofonia, G., Tzagaraki, E. and Alissandrakis, E. (2023). A low-cost, low-power, multisensory device and multivariable time series prediction for beehive health monitoring, *Sensors* **23**(3): 1407.

Semenoglou, A.-A., Spiliotis, E. and Assimakopoulos, V. (2023). Data augmentation for univariate time series forecasting with neural networks, *Pattern Recognition* **134**: 109132.

Teichgraeber, H. and Brandt, A. R. (2022). Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities, *Renewable and Sustainable Energy Reviews* **157**: 111984.

Time Series Decomposition (n.d.). Accessed: August 2023.
URL: *https://timeseriesreasoning.com/contents/time-series-decomposition*

Zhou, Y., Wang, S., Wu, T., Feng, L., Wu, W., Luo, J., Zhang, X. and Yan, N. (2022). For-backward lstm-based missing data reconstruction for time-series landsat images, *GIScience & Remote Sensing* **59**(1): 410–430.