information

Article

Time Series Forecasting Utilizing Automated Machine Learning (AutoML): A Comparative Analysis Study on Diverse Datasets

George Westergaard [1], Utku Erden [1], Omar Abdallah Mateo [1], Sullaiman Musah Lampo [1], Tahir Cetin Akinci [2,3] and Oguzhan Topsakal [1,*]

[1] Department of Computer Science, Florida Polytechnic University, Lakeland, FL 33805, USA
[2] Electrical Engineering Department, Istanbul Technical University, Istanbul 34467, Turkey
[3] Winston Chung Global Energy Center (WCGEC), University of California at Riverside (UCR), Riverside, CA 92521, USA
* Correspondence: otopsakal@floridapily.edu

Abstract: Automated Machine Learning (AutoML) tools are revolutionizing the field of machine learning by significantly reducing the need for deep computer science expertise. Designed to make ML more accessible, they enable users to build high-performing models without extensive technical knowledge. This study delves into these tools in the context of time series analysis, which is essential for forecasting future trends from historical data. We evaluate three prominent AutoML tools—AutoGluon, Auto-Sklearn, and PyCaret—across various metrics, employing diverse datasets that include Bitcoin and COVID-19 data. The results reveal that the performance of each tool is highly dependent on the specific dataset and its ability to manage the complexities of time series data. This thorough investigation not only demonstrates the strengths and limitations of each AutoML tool but also highlights the criticality of dataset-specific considerations in time series analysis. Offering valuable insights for both practitioners and researchers, this study emphasizes the ongoing need for research and development in this specialized area. It aims to serve as a reference for organizations dealing with time series datasets and a guiding framework for future academic research in enhancing the application of AutoML tools for time series forecasting and analysis.

Keywords: forecasting; time series; AutoML; machine learning; cryptocurrency; COVID-19; Bitcoin; weather; AutoGluon; Auto-Sklearn; PyCaret

Citation: Westergaard, G.; Erden, U.; Mateo, O.A.; Lampo, S.M.; Akinci, T.C.; Topsakal, O. Time Series Forecasting Utilizing Automated Machine Learning (AutoML): A Comparative Analysis Study on Diverse Datasets. *Information* **2024**, *15*, 39. https://doi.org/10.3390/info15010039

Academic Editor: Binbin Yong

Received: 20 November 2023
Revised: 7 January 2024
Accepted: 10 January 2024
Published: 11 January 2024

1. Introduction

Time series data are characterized as datasets aligned in a chronological sequence over a specified temporal interval [1]. Such data are frequently utilized in temporal analyses of events to discern trends or patterns, which may facilitate the forecasting or informed projection of future occurrences [2]. The incorporation of contextual data associated with the time series, such as seasonal variations or events that trigger fluctuations in metrics like sales, can further elucidate factors influencing the observed variables over time [3]. This comprehensive approach enables a more nuanced understanding of the dynamic interplay between the time series data and their external influences [4].

Given that time is one of the axes of time series datasets, they can become somewhat cumbersome to analyze manually, depending on the intervals at which the tracked data are measured [5]. Therefore, they would benefit from being analyzed through machine learning (ML) models. Machine learning approaches can significantly reduce the time needed to analyze large datasets [6]. Some problems that can be solved using time series analysis through machine learning are forecasting, anomaly detection, imputation, and classification [7]:

- Forecasting: Predicting future values of the time series data based on historical data from the same time series.

- Anomaly Detection: Identifying abnormal moments in the time series data that could possibly indicate that there is a problem with the data or room for improvement with the attribute being measured (i.e., selling jackets in the summertime).
- Imputation: Replace missing data in time series datasets with calculated values based on the trends of surrounding data.
- Classification: Prediction of categorical outcomes from the given time series data.

The use of ML algorithms for time series analysis has become increasingly important in a wide range of applications [7], from finance and economics to weather forecasting and industrial production [8]. However, building effective ML models for time series can be challenging [9], especially for non-experts who may not have the necessary knowledge of ML algorithms or programming skills [10].

In the field of machine learning, achieving the best performance often requires extensive knowledge, experience, and effort from humans [11]. One of the most challenging and time-consuming tasks is selecting the right tool for a specific dataset [12].

To address this challenge, AutoML has emerged as a set of techniques and frameworks that aim to automate the process of building and optimizing ML [13]. AutoML tools automate various steps involved in constructing ML models, including data preprocessing, feature engineering, model selection, and hyperparameter tuning. This automation helps to simplify the ML process. With simplification, it becomes easier and faster for non-experts to create effective ML models. AutoML tools also often integrate with popular programming languages and machine learning libraries, such as Python and TensorFlow, making them accessible to a wide range of users [14].

There are many types of AutoML tools, each with varying capabilities and features that allow them to solve different challenges in the ML pipeline [13]. Some of the most common capabilities of AutoML tools include data preparation, model selection, and hyperparameter tuning:

- Automated Data Preparation: These tools are designed to automate the process of cleaning and preparing data for machine learning [15]. This includes tasks such as data preprocessing, feature selection, and data augmentation. Automated data preparation tools can save time and reduce the risk of errors that can occur during manual data preparation [16].
- Automated Model Selection: These tools are designed to automatically select the best ML model for a given problem. They accomplish this by automatically comparing the performance of multiple models and selecting the one that performs the best on the given dataset. Automated model selection tools can save time and provide models with the highest tested accuracy [16].
- Automated Hyperparameter Tuning: Hyperparameters are parameters that the ML model cannot learn during the training [17]. They can be adjusted to improve the performance of ML models, but they must be tuned by the user, not the model itself [18]. Automated hyperparameter tuning tools are designed to automate the tedious task of finding the best hyperparameters for a given model. This can save time and improve the accuracy of ML models [19].

AutoML has recently seen significant advancements as well as increased popularity amongst the public, leading to a demand for automated systems that can support both experts and novices in creating ML applications quickly and efficiently [20].

Certain AutoML tools are engineered to automate the complete ML pipeline, encompassing stages from data preparation to model deployment, proving beneficial for entities lacking a specialized ML team or the expertise to construct and deploy these models [21]. These tools offer a rapid pathway for prototyping models or concepts, allowing for exploration prior to a significant investment of time and resources [22]. This automation facilitates efficiency and accessibility in ML processes, especially for organizations with limited ML capabilities [23].

There are plenty of AutoML tools available for public use, either free or paid [24]. As with any tool, the one you choose can specialize in specific fields of ML approaches and

models [25]. Since this study is looking into evaluating how the AutoML tools function when trained for time series data, the options were narrowed to choose the following AutoML tools that are better equipped for time series: H_2O ai [26], TPOT [27], AutoTS [28], MLJAR [29], Kedro [30], AutoGluon [31], Azure Machine Learning [32], Amazon Forecast [33], PyCaret [34].

Given the large number of Automated Machine Learning tools available for time series analysis, it is important to conduct a rigorous comparison and evaluation of these different tools to discern their merits and limitations. The main objective of this paper is to comprehensively evaluate the leading AutoML tools used in the field of time series analysis. We quantify the adequacy of these tools by subjecting them to scrutiny using three datasets characterized by different properties while employing a large number of performance metrics. By undertaking this endeavor, our goal is to provide valuable insights into optimal strategies for the judicious selection and effective use of AutoML tools tailored to the requirements of time series analysis.

Time series data are a ubiquitous component of modern-day analysis in numerous fields, including finance, economics, meteorology, and traffic management. Analyzing these data and making informed forecasts is of significant importance. Although various models, including linear modeling and deep learning, are utilized for predictive forecasting, AutoML frameworks have attracted considerable attention due to their ability to autonomously identify and optimize models. Our investigation seeks to contribute to the existing literature on AutoML for time series data analysis by providing contemporary insights and analyses on the topic.

2. Related Works

In the "Review of ML and AutoML Solutions to Forecast Time-Series Data" [12], the different methods used to analyze time series data for forecasting, including linear modeling, deep learning, and AutoML frameworks, are reviewed and analyzed. It highlights the growing importance of time series analysis and identifies the challenges associated with modeling this type of data. While linear models are the simplest, they may not provide the highest prediction accuracy. In contrast, deep learning models require expertise and patience to design and optimize. AutoML frameworks can automatically find and optimize ML models for time series data, but they are still evolving and require further research. The article provides a comparison of different AutoML frameworks and techniques and suggests that further empirical studies are needed to compare and evaluate their potential. This closely resembles what we are attempting to achieve with this paper, but it is more focused on forecasting rather than varied datasets.

In this article [12], a comprehensive examination is undertaken of various methodologies employed in predictive forecasting. The review underscores the escalating significance of time series analysis and highlights the complexities associated with modeling this specific genre of data. It encompasses an exploration of linear modeling, deep learning, and AutoML frameworks. Although linear models may exhibit simplicity, their predictive accuracy might not be optimal. On the other hand, deep learning models necessitate expertise and persistence in their design and optimization. AutoML frameworks are capable of autonomously identifying and refining ML models for time series data; however, they are in the developmental phase and require further research.

Our research is closely aligned with prior studies. Still, it differs in that we emphasize the investigation of AutoML frameworks across various datasets rather than focusing solely on predictive forecasting. Our findings include a comparative analysis of disparate AutoML frameworks and techniques, concurrently highlighting the necessity of empirical research to evaluate their prospective efficacy.

"Time Series Data Modeling Using Advanced Machine Learning and AutoML" [13] discusses the importance of modeling a time series dataset using past data and explores the use of machine learning, deep learning, and AutoML methods. It focuses on the data drift problem, the change in input data to the model that leads to worse performance,

and uses historical time series data from the cryptocurrency market to compare the efficiency of different techniques. The results indicate that while AutoML is in its early stages of development, the demonstrated methods can be used as a starting point for accurate modeling. The study suggests future work should include designing a new AutoML framework for time-series forecasting and conducting comparative studies using appropriate performance metrics.

The paper "Review of automated time series forecasting pipelines" [34] discusses the increasing demand for time series forecasting and the need for automation to make the design process more efficient. It reviews existing literature on automated time series forecasting pipelines and analyzes how each of the five sections of pipeline structure (data preprocessing, feature engineering, hyperparameter optimization, forecasting method selection, and forecast ensembling) are automated, incorporating both AutoML and automated statistical forecasting methods. The article finds that most of the reviewed papers only cover two or three of the five sections of the forecasting pipeline, indicating a research gap regarding approaches that holistically consider the automation of the forecasting pipeline. The article suggests future work should research the adaption of the identified automation methods for probabilistic time series forecasts, validate and tailor the automated forecasting pipelines to particular use cases, and examine their performance in terms of forecast accuracy and computing effort. Overall, the article highlights the need for open-source publishing to promote the adoption of automated pipelines for time series forecasting and suggests considering the automated application of resulting forecasting models to fully automate the entire forecasting process.

3. Methodology

To evaluate AutoML tools for time series, we reviewed the available AutoML tools for public use and selected three. In the following sections, we first introduce the selected AutoML tools and our selection criteria. We then introduce the selected datasets and their properties. Following this, we review the metrics that will be used to numerically evaluate the tools. Finally, we discuss the methods for utilizing the tools and how they are prepared.

3.1. AutoML Tools to Be Evaluated

Our criteria for choosing the AutoML tools include whether they can handle time series data as well as their popularity, performance, ease of use, open-source nature, and active development. We present the details of the selected three AutoML tools below:

AutoGluon:

AutoGluon automates many of the tasks in machine learning, including data preparation, feature engineering, model selection, and hyperparameter selection using advanced tuning algorithms [35]. With AutoGluon, users can achieve state-of-the-art performance in image classification, object detection, text classification, and traditional tasks like classification and regression. Experts can also customize the process by specifying certain hyperparameter ranges or using AutoGluon to tune their own custom models. Additionally, AutoGluon can distribute its computation across multiple machines to return trained models more quickly.

AutoGluon fits various models ranging from off-the-shelf boosted trees to customized neural networks. The models are combined in a distinctive approach where they are stacked in various layers and trained layer by layer. This technique guarantees that the unprocessed data can be converted into accurate and reliable predictions within a specific time limit.

Auto-Sklearn 2.0:

Auto-Sklearn 2.0 is an open-source library for Automated Machine Learning that builds upon the success of its predecessor, Auto-Sklearn. It offers a powerful framework for solving a wide range of ML problems, including classification, regression, and time

series forecasting. It provides a range of preprocessing techniques, feature selection, and model selection strategies, including deep learning models.

Auto-Sklearn 2.0 also includes state-of-the-art ensemble methods, such as stacking and blending, that improve the accuracy of predictions. These features make Auto-Sklearn 2.0 a valuable tool for practitioners and researchers alike, as it simplifies the process of ML by automating many of the manual steps required for building models. Additionally, Auto-Sklearn 2.0 offers significant performance improvements over manual methods, which can save time and resources [36].

PyCaret:

PyCaret is a Python-based ML library that is available as an open-source library and facilitates low-code development. This library automates machine learning workflows and serves as an end-to-end machine learning and model management tool. With PyCaret, users can increase productivity by quickly experimenting with different algorithms and settings, compare the performance of different models, and generate insights from the data without requiring a deep understanding of ML theory or programming [37]. PyCaret can automatically perform many of the tedious and time-consuming tasks involved in machine learning, such as data preprocessing, feature selection, model selection, and hyperparameter tuning. Overall, PyCaret's AutoML capabilities help abstract machine learning, making it more accessible and usable for a wider range of users [38].

The similarities and differences between AutoGluon, Auto-Sklearn 2.0, and PyCaret in several key areas are listed in Table 1 and are further explained as follows:

- *Flexibility:* AutoGluon and PyCaret support a wider range of ML tasks and algorithms, while Auto-Sklearn 2.0 focuses on classification and regression tasks only. AutoGluon also provides more flexibility in terms of customization and user-defined constraints.
- *Ease of Use:* All three frameworks are designed to be easy to use and require minimal programming knowledge. AutoGluon and PyCaret are designed to be accessible to non-experts, with user friendly interfaces and tutorials.
- *Methodology:* AutoGluon and Auto-Sklearn 2.0 are designed to maximize the performance of ML models, using a combination of meta-learning, ensemble methods, and other techniques. PyCaret is designed to be a high-performance library that can quickly and efficiently process large datasets and build accurate ML models.
- *Open Source:* All three frameworks are open source, which means that they are freely available for anyone to use and modify. AutoGluon, Auto-Sklearn 2.0, and PyCaret are developed by researchers.
- *Community:* All three frameworks have active communities of developers and users, with resources such as tutorials, documentation, and forums available to users.

Table 1. Comparison of AutoML frameworks: AutoGluon, Auto-Sklearn, and PyCaret.

Model	***AutoGluon***	***Auto-Sklearn***	***PyCaret***
Flexibility	Support a wider range of tasks and algorithms	Focuses on classification and regression tasks	Support a wider range of tasks and algorithms
Ease of Use	minimal programming knowledge; designed to be accessible to non-experts	minimal programming knowledge	minimal programming knowledge; designed to be accessible to non-experts
Methodology	uses a combination of meta-learning and ensemble methods	quickly and efficiently process large datasets	uses a combination of meta-learning and ensemble methods
Open Source	Yes	Yes	Yes
Community	Active	Active	Active

In summary, while AutoGluon, Auto-Sklearn 2.0, and PyCaret share many similarities as they are all AutoML frameworks, they also have their unique strengths and weaknesses. Depending on the specific needs of the user, one framework may be more suitable than the others.

3.2. Datasets

We selected three types of time-series datasets, Bitcoin, COVID-19, and weather, which present different challenges for the selected AutoML tools. We explain the properties of these datasets below:

Bitcoin Dataset:

The highly volatile and variable nature of Bitcoin prices demanded an exceptional analytical tool that could disentangle complex historical patterns and prognosticate future trends. AutoML tools provide a fast, precise, and efficient approach to analyzing the intricate and inherently uncertain nature of Bitcoin data.

AutoML technology promises exceptional efficacy in dissecting historical data and making precise projections about Bitcoin's future value and helps uncover previously hidden, latent patterns that are crucial in decision-making processes to acquire, divest, or retain Bitcoin assets.

In the study of Bitcoin data, it is important to be cautious and consider all relevant factors when conducting analyses or making predictions. This involves using data from multiple sources, including cryptocurrency market news, trading volume, investor attitudes, and broader economic factors such as interest rates and inflation rates. The time series analysis showed significant patterns in Bitcoin's value over time, notably initial increases that have since leveled off. Our research indicates that using an AutoML-based method improves the accuracy and usefulness of our forecasts, offering valuable insights into upcoming trends.

We based our predictions on a training dataset of 2509 instances, with a testing dataset comprising 280 instances. Our principal objective was to anticipate 'adj close' values, representing a holistic predictive endeavor. Figure 1 plots the values of the Adjacent Close between 2014 and 2022. Our analysis of Bitcoin's price history revealed intricate patterns that could inform sound investment decisions. By leveraging AutoML technology and exercising judicious circumspection in incorporating all pertinent determinants, it is possible to generate insights that would be instrumental in optimizing the investment portfolios in Bitcoin and other cryptocurrencies.

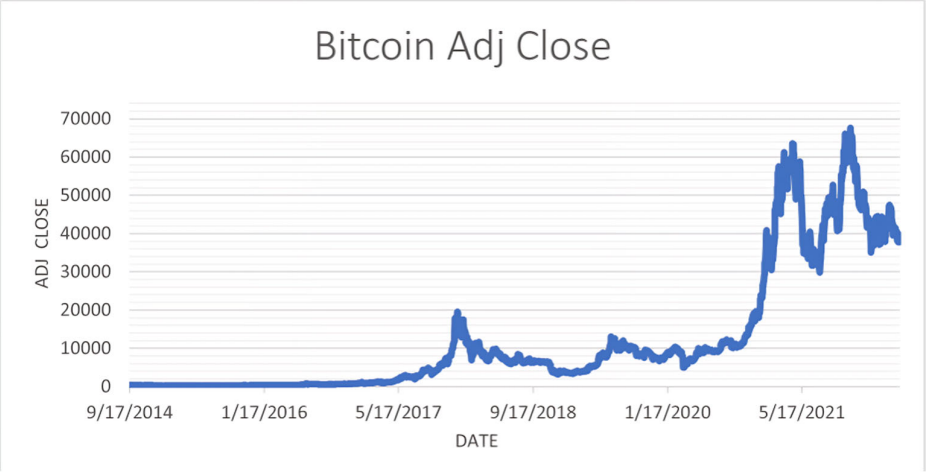

Figure 1. Bitcoin Adjacent Close from 2014 to 2022.

COVID-19 Dataset (Delhi):

Delhi's COVID-19 dataset is suitable for time series analysis due to its impact on the public, temporal trends, non-stationarity, policy implications, and predictive power. Time series analysis can help understand how the virus spreads, the impact of interventions and vaccination efforts, and inform future policy decisions. It can also be used to forecast future case counts, hospitalizations, and deaths for public health planning and resource

allocation. Figure 2 displays the number of deaths in Delhi during the COVID-19 pandemic. The training data include 216 rows, and the test data include 24 rows.

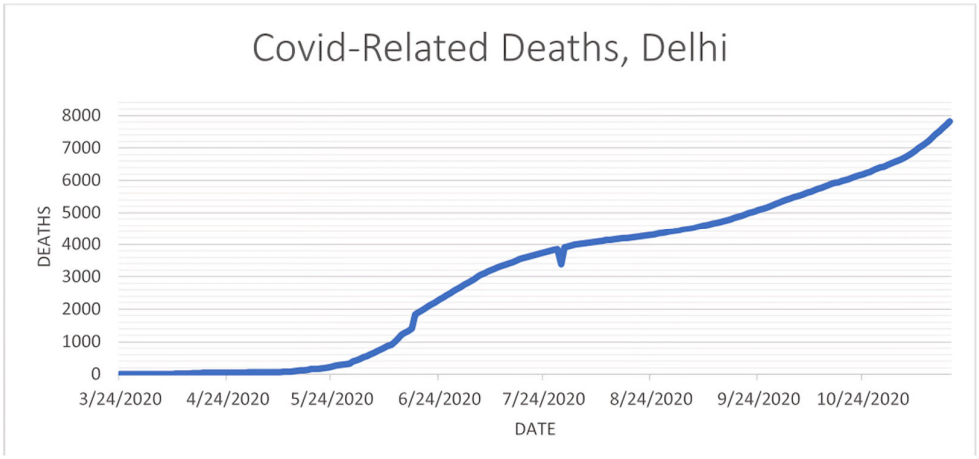

Figure 2. Delhi deaths from COVID-19 over time in 2020.

In the dataset, there is an unexpected decrease in reported COVID-19-related deaths in Delhi on 24 July 2020. The exact cause of this dip is uncertain, but it may be attributed to factors such as reporting delays, health policy changes, or updates in data collection methods. Such anomalies underscore the complexities of interpreting time-series health data and highlight the need for careful analysis. The potential reasons provided here need further investigation and emphasize the dynamic nature of pandemic data reporting.

The COVID-19 dataset is too small to have a good prediction; however, that is the problem for many cases, so it is interesting to see how AutoML tools will perform with this dataset. The dataset has multiple columns; however, for this experiment, 'Date' is used as the index, and 'Deaths' is used to predict the number of deaths. The selection of the COVID-19 dataset was based on its relatively low number of rows and consistent pattern of data. This dataset never had any decrease in deaths; it was a cumulative total and not just a day-to-day count.

Weather Dataset (Delhi):

We selected the weather change dataset for time series analysis based on the availability and reliability of the dataset, the significance of weather change as a global issue, and its possibility of obtaining insights about the weather dataset patterns. The weather dataset is the most stable dataset compared to the other datasets used in this study. The weather has similar outcomes throughout the years with a small variation, which makes it easier to predict what the temperature will be over time. The training data include 1315 rows, and the test data include 148 rows. The objective is to forecast the "average temperature" values. Figure 3 presents a plot of the datasets over several years.

The weather dataset is a valuable source of information for time series analysis for several reasons:

- *Temporal Dependence:* Time series analysis is well-suited for analyzing data with a temporal dependence, i.e., how much previous data influence the future data. Such dependence can be seen in the weather change dataset, which records changes in temperature, pressure, humidity, precipitation, and other weather variables over time.
- *Short-term forecasting:* Time series analysis can be used to forecast future weather patterns, such as temperature, rainfall, or snowfall, which can be used to inform short-term planning and decision-making, such as in agricultural production or transportation logistics.
- *Seasonality:* The weather change dataset exhibits strong seasonality, with cyclic patterns of variation across different seasons, months, and years. Time series analysis can be

used to identify and model these patterns, which can be used to make more accurate predictions about future weather trends.

- *Trend Analysis:* Time series analysis can be used to identify long-term trends in the weather change dataset, such as increasing global temperatures or changes in precipitation patterns.

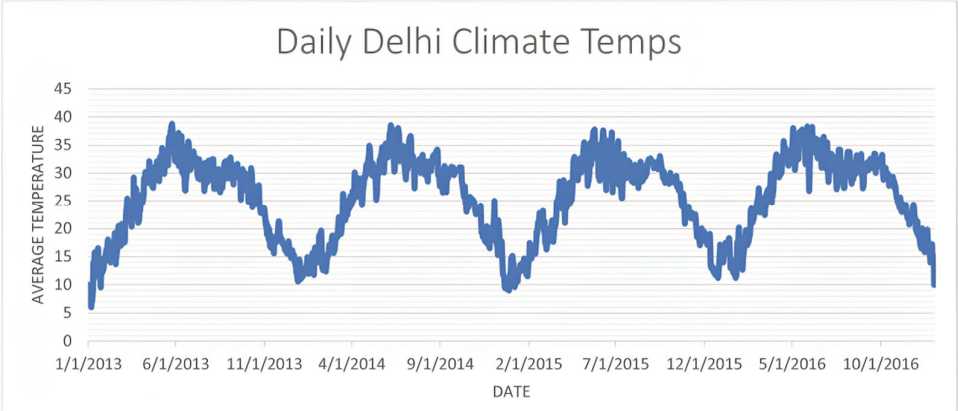

Figure 3. The average temperature in Delhi, India, over several years.

3.3. Comparing Statistics of Datasets

Table 2 provides the descriptive statistics, such as standard deviation, mean, skewness, and kurtosis, for the target labels of the three datasets.

Table 2. Descriptive statistics on datasets.

Dataset	*Std. Dev.*	*Mean*	*Skewness*	*Kurtosis*
COVID-19	2406.22	3109.37	−0.001	−1.332
Bitcoin	16,615.38	12,126.41	1.65	1.47
Weather	7.54	9.38	0.62	−0.33

Statistical analysis of the COVID-19 dataset is challenging since the data only increase without any possible decrease. Since nobody can come back to life, deaths are only counted when someone has died; therefore, skew and kurtosis do not work as a faithful data format over time as opposed to deaths per day.

The Bitcoin dataset shows a much larger standard deviation (16,615.381) and a higher mean (12,126.41) than the COVID-19 dataset. The positive skewness value of 1.65 indicates that the distribution is right-skewed, with a long tail to the right, and the kurtosis value of 1.47 suggests that it has a higher peak and heavier tails than a normal distribution, which is a leptokurtic distribution.

The weather dataset has the smallest standard deviation (7.54) of the three sets and a mean (9.38) that is closer to the mode than the previous sets. The positive skewness value of 0.62 indicates that the data are moderately right-skewed, and the negative kurtosis value of −0.33 suggests that it has flatter tails and a slightly flatter peak than a normal distribution. This is not completely accurate, which shows that some statistics are not always able to generalize data, especially if the data do not act in a smooth manner or when they have sinusoidal patterns.

Overall, these values describe different types of distributions with varying degrees of symmetry, kurtosis, and skewness, even though the deaths are not the correct data type on which these metrics should be used. The interpretation of these values can provide insights into the nature of the data and can be used to make inferences about the underlying patterns or trends that an ML model would also pick up on.

3.4. Metrics to Evaluate AutoML Tools

There are several metrics that can be used to evaluate the performance of ML tools. Every metric is used depending on the type of problem or dataset. We have utilized the following metrics to evaluate the performance of AutoML tools on time series data. In these metrics, n is the number of samples, yi is the actual value, and ŷi is the predicted value.

Mean Absolute Error (MAE) measures the average absolute difference between the predicted and actual values, in this case, weather variables such as temperature, humidity, or precipitation. It can provide a good overall sense of how well the model can predict the weather variables.

$$MAE = \left(\frac{1}{n}\right) * \sum |yi - \hat{y}\mathrm{i}| \tag{1}$$

Root Mean Squared Error (RMSE) measures the average squared difference between the predicted and actual values and is the square root of the MSE. RMSE is a commonly used metric in weather forecasting since it heavily penalizes large prediction errors.

$$RMSE = \sqrt{\left[\left(\frac{1}{n}\right) * \sum (yi - \hat{y}\mathrm{i})^2\right]} \tag{2}$$

Mean Absolute Percentage Error (MAPE) measures the percentage difference between the predicted and actual values. It can provide insight into the accuracy of the model's predictions relative to the actual values.

$$MAPE = \left(\frac{1}{n}\right) * \sum |(yi - \hat{y}i)/yi| * 100\% \tag{3}$$

Mean Absolute Scaled Error (MASE) is a metric used to evaluate the accuracy of time series forecasting models. MASE compares the performance of a model with the performance of a naive model, which is a simple method that uses historical values to make predictions.

$$MASE = \frac{\left(\frac{1}{n}\right)\sum |(\mathrm{yi} - \hat{\mathrm{y}}\mathrm{i})|}{\left(\frac{1}{n-1}\right)\sum |\mathrm{yi} - \mathrm{y}\{\mathrm{i} - 1\}|} \tag{4}$$

3.5. Steps to Use the AutoML Tools

We utilized Google Colab for benchmarking all three AutoML tools to ensure fair computing power. The steps and the methodology followed during the evaluation of the AutoML tools are described next.

AutoGluon:

AutoGluon (version 0.7.0), like other AutoML software, has specific requirements for data structure and presentation of its predictor functions. The data should consist of at least three columns: a unique ID (integer or string), a compatible timestamp indicating the analyzed value's time, and the numeric value of the time series. Although the column names are not crucial, they should be specified when constructing a 'Time Series Data Frame' for AutoGluon's prediction. Failure to match the required format will result in an exception and the failure to execute. Table 3 presents an example of the correct dataset format.

Table 3. AutoGluon data format requirement.

Item_id	Date	Deaths
H1	24 March 2020	0
H1	25 March 2020	1
H1	26 March 2020	1
H1	27 March 2020	2
...	...	...

To create the necessary data structure 'TimeSeriesDataFrames' for AutoGluon, the dataset must be loaded into pandas dataframes for both training and testing. The 'TimeSeriesDataFrame.from_data_frame' function is utilized to construct a 'TimeSeriesDataFrame' by specifying parameters such as the data frame, ID column name, and timestamp column name. The same format is employed for the conversion of test data.

AutoGluon employs 'TimeSeriesPredictor' objects to forecast future values of time series data. These predictor objects have essential parameters for customizing the forecasting model, including the prediction length, the path for saving trained models, the target column to track, and the evaluation metric used for model fitting. The 'fit()' function requires parameters such as the 'TimeSeriesDataFrame' of the training data, a quality preset, and a preferred time limit.

The quality preset impacts model complexity and training time in AutoGluon fitting. Dataset size and hardware also influence training time. The 'fit()' function takes a time limit parameter for maximum fitting duration. It guides AutoGluon to avoid new fittings exceeding the remaining time. Without a time limit, all models for the specified model preset are trained. AutoGluon outputs trained models, fitting duration, best model, and its score.

Following the model fitting process, the AutoGluon predictor generates future time series forecasts for the provided dataset. By default, it utilizes the model with the highest score on the validation set. Forecasts initiate from the end of the training data, aligning with the specified number of predictions. The 'predict()' function requires the 'TimeSeriesDataFrame' training dataset as input and delivers predictions in a corresponding format. The forecasts are probabilistic, encompassing a range of outcomes spanning a 10% to 90% likelihood. The mean, signifying the expected value, emerges as the most influential outcome, denoting the average likelihood.

AutoGluon offers several presets, allowing for the easy comparison of its models. Higher-quality presets generally yield more accurate predictions but require longer training times and result in less computationally efficient models. After generating the forecasted values, AutoGluon offers visualization and evaluation tools for analysis.

AutoGluon's leaderboard() function allows the evaluation of each trained model's performance by displaying test scores, which compare predictions to the test data, and validation scores derived from an internal validation set.

Auto-Sklearn:

There is no need to manipulate the dataset when using Auto-Sklearn to train a model, as the platform automatically handles the organization of datetime values. During the pre-training phase, only the date and label values are selected. It is crucial to specify both the training and testing datasets and to separate the label from the dataset. The training set comprises the first 90% of the data, while the test set contains the remaining 10%.

During the training, we utilized autosklearn.regression class's AutoSklearnRegressor() method using the default parameters, with the exception of 'time_left_for_this_task' and 'n_jobs'. The former parameter defines the time_limit in seconds for searching suitable models while increasing its value enhances the likelihood of obtaining better models. The latter parameter specifies the number of jobs to run in parallel for fit (), where -1 denotes the utilization of all processors. 'n_jobs' was utilized to expedite the process, while 'time_left_for_this_task' was used to monitor how the training success depends on the duration of the model search. The remaining parameters were kept as default values, given that the purpose was to observe how the tool handles the data with its default settings.

PyCaret:

PyCaret offers well-documented tutorials and examples for various scenarios. However, it requires users to have some knowledge of ML to prepare the dataset in the required format. PyCaret provides several functions that simplify the forecasting process. The following functions were used: (1) initialization: setup(), (2) training: compare_models(), (3) analysis: plot_model() and evaluate_model().

The setup() function in PyCaret allows customization of the preprocessing pipeline and ML workflow. It requires two parameters: the dataset and the target variable. It must be called before executing other functions.

PyCaret trains multiple models and ranks them from best to worst based on performance metrics. The compare_models() function outputs the models organized by their scores, which can take varying amounts of time depending on the dataset size and type.

PyCaret includes the plot_model() function, which allows for visualization of the performance of trained ML models. It provides graphical representations of performance metrics and diagnostics, aiding in the interpretation and understanding of the model's behavior.

4. Results

All AutoML tools were tested using a 90/10 percent training/test dataset split, and the RMSE, MASE, and MAPE metrics were used for the evaluation.

AutoGluon Results:

We provide graphs for each trained preset, and the prediction scores are displayed in tabular form as they are printed by AutoGluon. These results showcase the top three models trained by AutoGluon while also listing all the trained models to highlight the differences between AutoGluon's quality presets.

Bitcoin Data:

Accurately predicting speculative things like Bitcoin's price is very challenging. However, it is possible to develop a model that assists in making predictions. We list the top three models generated by AutoGluon and their corresponding RMSE, MASE, and MAPE scores for the Bitcoin data in Table 4.

Table 4. AutoGluon best_quality score results for RMSE, MASE, and MAPE on Bitcoin dataset.

Model	**RMSE**	*Model*	**MASE**	*Model*	**MAPE**
DeepAR/T3	−8461.32	*WeightedEnsemble*	−151.27	*WeightedEnsemble*	−0.23
WeightedEnsemble	−8535.61	*DeepAR*	−151.95	*DeepAR/T1*	−0.26
DeepAR/T6	−10,870.43	*SimpleFeedForward*	−176.07	*SimpleFeedForward/T1*	−0.52

Figure 4 displays the past and future values from a point in the dataset, along with the mean forecast (predictions) and the confidence interval of the values when the MAPE metric is used. Even with AutoGluon's 'best_quality' setting, the AutoML tool was unable to accurately predict the peaks and troughs that come with erratic stock prices. In fact, the confidence levels appear to have very large gaps to show that the predictions are just as unknown to the tool as they are to its human counterpart.

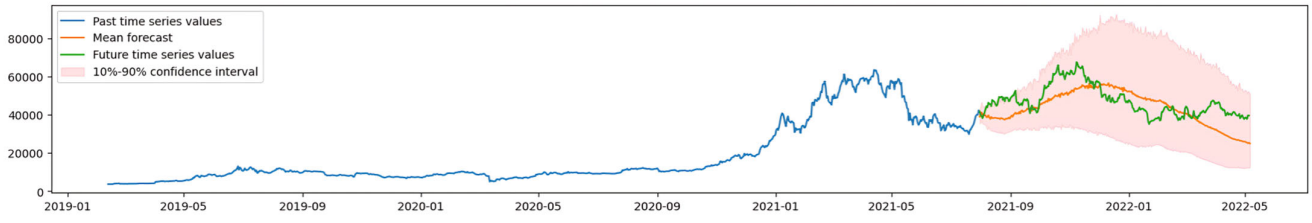

Figure 4. AutoGluon tool predictions with the 'best quality' setting using the MAPE metric on the Bitcoin dataset.

Weather Data: AutoGluon can recognize the seasonal pattern of the weather data and closely match it, as seen in Figure 5. The repeated pattern of weather data can then be understood and replicated by the model. While it is not perfect, the predictions carry a similarity that matches the test data. The top three models and their RMSE, MASE, and MAPE scores for the Weather data are in Table 5.

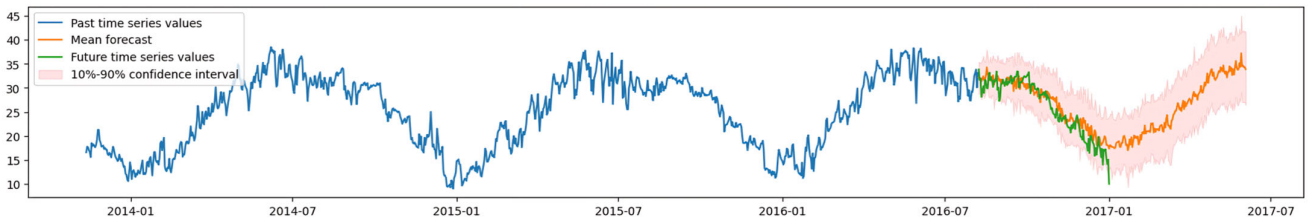

Figure 5. AutoGluon tool predictions trained with the 'best quality' setting using the MAPE metric on the weather dataset.

Table 5. AutoGluon best_quality score results for RMSE, MASE, and MAPE on weather dataset.

Model	RMSE	*Model*	MASE	*Model*	MAPE
WeightedEnsemble	−2.49	*WeightedEnsemble*	−1.91	*WeightedEnsemble*	−0.09
TemporalFusionTransformer	−2.94	*TemporalFusionTransformer*	−2.11	*TemporalFusionTransformer*	−0.11
DeepAR/T2	−4.90	*DeepAR/T1*	−2.35	*SimpleFeedForward/T2*	−0.14

COVID-19 Data:

Even with the peculiarity of the ever-rising data, AutoGluon recognized a simple, steady increase in the data. With historical data, it is difficult to predict exponential growth, like the small bit in the test data shown. This is not necessarily the fault of the AutoML tool, but it can also show that predictions for data that can suddenly turn into a completely different algorithmic function are nearly impossible without large amounts of data on which to base said predictions. The top three models generated by AutoGluon are listed for RMSE, MASE, and MAPE scores for the Bitcoin data in Table 6. Figure 6 displays the predictions and the confidence interval of the values when the MAPE metric is used.

Table 6. COVID-19 deaths—AutoGluon best_quality score results for RMSE, MASE, and MAPE.

Model	RMSE	*Model*	MASE	*Model*	MAPE
WeightedEnsemble	−137.10	*WeightedEnsemble*	−1.15	*WeightedEnsemble*	−0.01
Theta/T2	−502.53	*DynamicOptimizedTheta*	−1.16	*DynamicOptimizedTheta*	−0.01
Theta/T1	−502.55	*ARIMA*	−4.23	*ARIMA*	−0.02

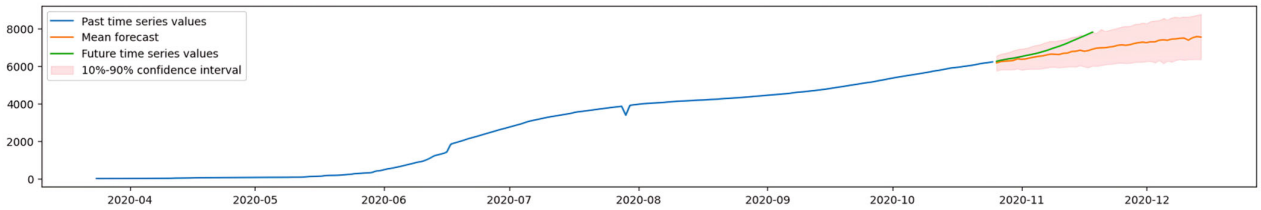

Figure 6. AutoGluon tool prediction trained with the 'best quality' setting using the MAPE metric on the COVID-19 dataset.

Auto-Sklearn Results:

The training time parameter for the Auto-Sklearn models was set to 3600 s to ensure comparability between models. The selection of the best model was based on the RMSE decision metric.

Bitcoin Data: Table 7 displays the best-performing model and its corresponding scores for training 3600 s (1 h).

Table 7. Bitcoin—AutoSklearn results for RMSE, MAPE, and MAE.

Model	*Ensemble Weight*	*RMSE*	*MAE*	*MAPE*
Gradient_boosting	0.073	22,999.52	19,760.62	0.44
Gradient_boosting	0.089	25,435.26	22,354.22	0.50
Gradient_boosting	0.09	31,333.02	25,987.67	0.56

Figure 7 shows the real test values and the predictions of the best model. Due to the lack of a proper distribution of the Bitcoin data over time, predicting it with high accuracy is challenging for the model.

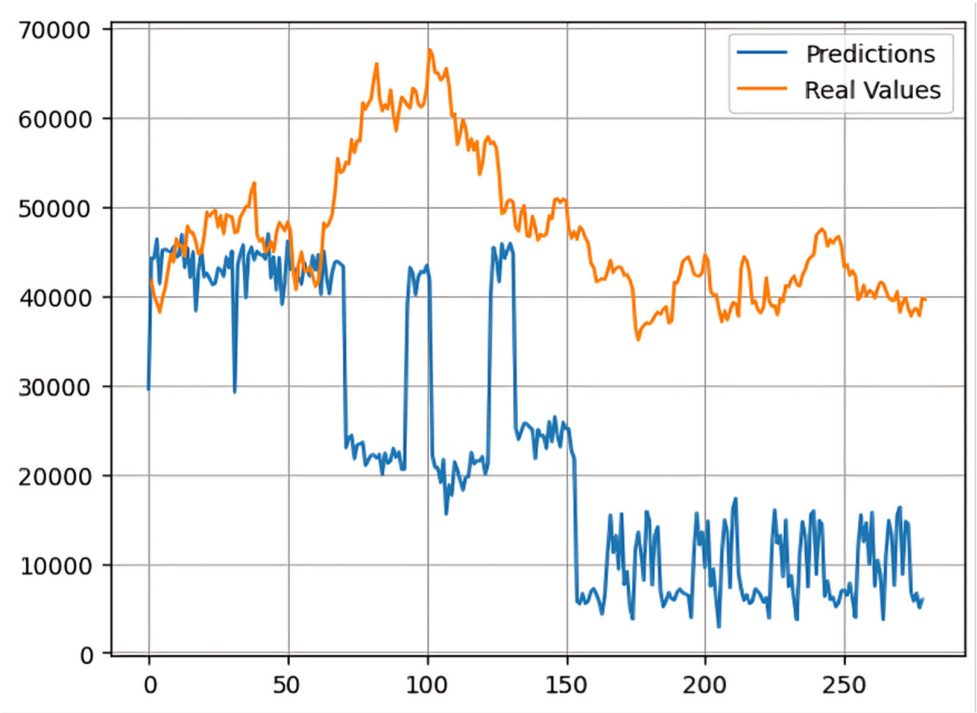

Figure 7. Auto-Sklearn tool predictions for the Bitcoin dataset.

Weather Data: Table 8 presents the selected model and its corresponding scores for the weather dataset. The weather data exhibit a discernible pattern, which allows for more accurate predictions compared to the Bitcoin dataset. Figure 8 displays the predicted and real values for the weather dataset.

Table 8. Weather—AutoSklearn results for RMSE, MAPE, and MAE.

Model	*Ensemble Weight*	*RMSE*	*MAE*	*MAPE*
Gradient_boosting	0.28	2.07	1.68	0.07
Gradient_boosting	0.12	2.56	2.01	0.13
Gradient_boosting	0.18	2.88	2.32	0.16

COVID-19 Data:

Table 9 summarizes the top-performing models in the existing literature, while Figure 9 graphically illustrates the actual and predicted values for the COVID-19 dataset. Despite similarities in pattern to the weather dataset, prediction accuracy for the COVID-19 data is consistently lower across all time samples. This discrepancy is attributed to the smaller sample size in the COVID-19 dataset and Auto-Sklearn's requirement for a minimum sample threshold to achieve high forecast accuracy.

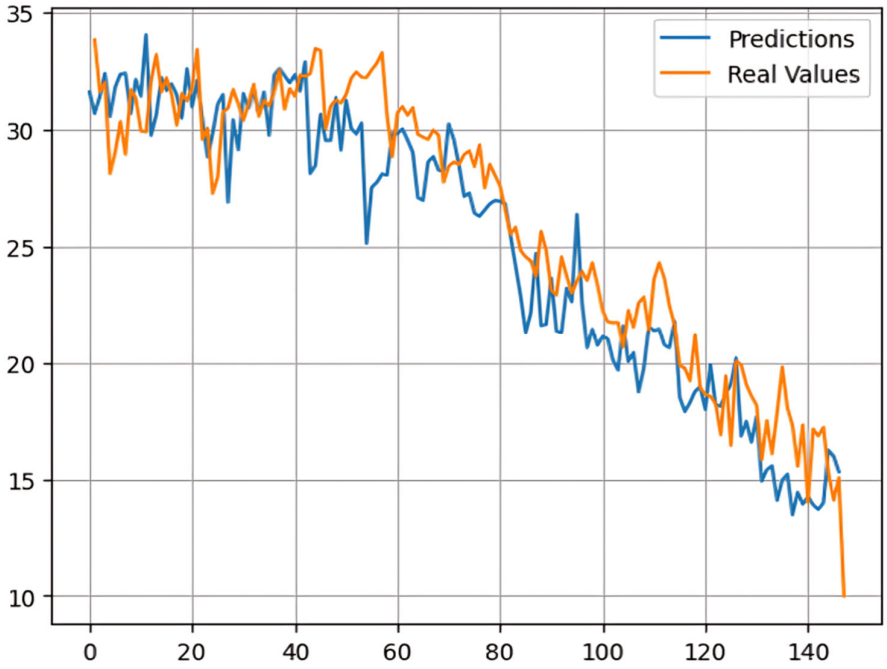

Figure 8. Auto-Sklearn tool predictions for the weather dataset.

Table 9. Auto-Sklearn results on COVID-19 dataset.

Model	*Ensemble Weight*	*RMSE*	*MAE*	*MAPE*
Libsvm_svr	0.90	1938.71	3345.30	0.47
Adaboost	0.08	3556.45	3823.22	0.55
Adaboost	0.02	4129.23	4022.32	0.61

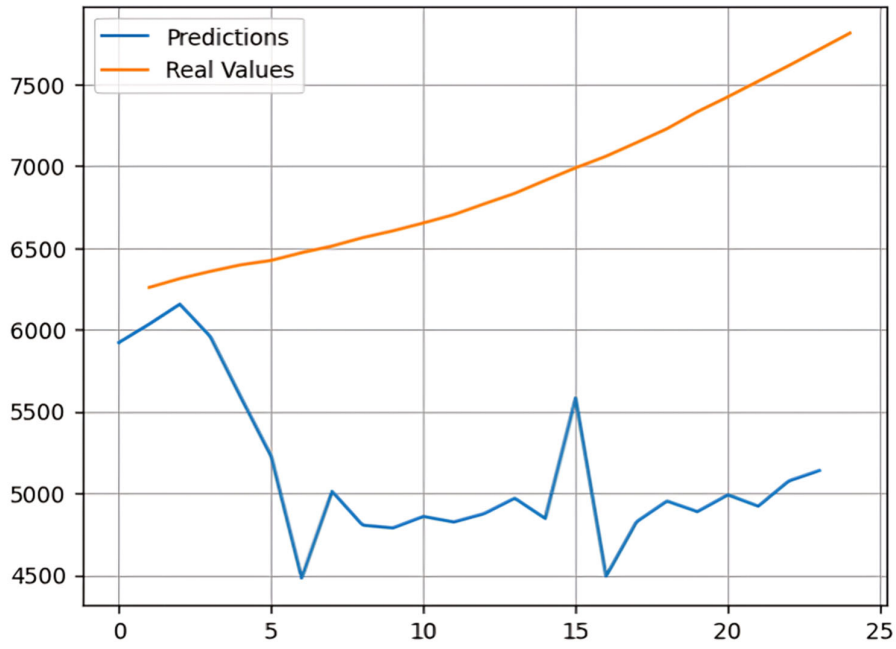

Figure 9. Auto-Sklearn tool predictions for the COVID-19 dataset.

PyCaret Results:

For the PyCaret tool, there is no need to split the data into train and test; however, it has been used as a function to obtain the number that equals the 10% of entries of the

dataset as a reference. PyCaret evaluates 121 different models to find the best-performing models. Tables 10–12 show the five top models using the Bitcoin, weather, and COVID-19 datasets, respectively.

Table 10. PyCaret results for MASE, RMSSE, MAE, RMSE, MAPE, SMAPE, and R2 metrics on the Bitcoin dataset.

Model	*MASE*	*RMSSE*	*MAE*	*RMSE*	*MAPE*	*SMAPE*	R^2
Decision Tree	8.4349	5.27	8760.98	10,311.07	0.32	0.42	−1.80
Light Gradient Boosting	8.6752	5.43	9048.89	10,628.55	0.32	0.42	−1.69
ARIMA	8.7116	5.43	9048.89	10,604.70	0.32	0.42	1.65
Extreme Gradient Boosting	8.7425	5.46	9103.50	10,679.71	0.33	0.43	−1.84
Gradient Boosting	9.0367	5.5965	9417.6893	10,952.0583	0.3373	0.4497	−1.9221

Table 11. PyCaret results for MASE, RMSSE, MAE, RMSE, MAPE, SMAPE, and R2 metrics on the weather dataset.

Model	*MASE*	*RMSSE*	*MAE*	*RMSE*	*MAPE*	*SMAPE*	*R2*
Extra Tree	0.60	0.57	2.49	3.08	0.10	0.10	0.10
K Neighbors	0.64	0.66	2.90	3.56	0.12	0.12	−0.17
Random Forest	1.22	1.15	5.48	6.16	0.28	0.22	−2.34
Linear	1.32	1.24	5.93	6.68	0.28	0.23	−2.78
Ridge	1.32	1.24	5.93	6.68	0.28	0.23	−2.78

Table 12. PyCaret results for MASE, RMSSE, MAE, RMSE, MAPE, SMAPE, and R2 metrics on the COVID-19 dataset.

Model	*MASE*	*RMSSE*	*MAE*	*RMSE*	*MAPE*	*SMAPE*	R^2
Orthogonal Matching Pursuit	0.0356	0.0345	51.6159	60.2779	0.0103	0.0104	0.7877
Huber	0.0423	0.0402	62.2791	71.4915	0.0129	0.0127	0.6205
Bayesian	0.0438	0.0429	64.1838	75.7905	0.0132	0.0133	0.6216
Exponential Smoothing	0.0500	0.0561	71.7088	97.0562	0.0141	0.0142	0.7424
Lasso	0.0597	0.0690	87.4104	121.7145	0.0181	0.0182	0.3064

5. Discussion

As technology evolves, new methodologies and tools for predictive forecasting using time series data will continue to emerge. AutoML frameworks are among the most promising methodologies that provide considerable potential for automated data processing and the optimization of predictive models. The incorporation of contemporary frameworks with an empirical study of their efficacy has significant implications for time series data analysis, predictive forecasting, and other associated fields. Next, we continue our discussion focusing on each dataset.

Bitcoin dataset:

The reason why each tool produces different scores can be attributed to several factors, such as the following:

Algorithm selection: Each tool uses different algorithms to train models on the given dataset. For example, PyCaret uses a decision tree regressor, while Auto-Sklearn 2.0 and AutoGluon use gradient boosting and ensemble models, respectively. Different algorithms can have varying strengths and weaknesses and may be better suited for different types of data.

Hyperparameter tuning: Each algorithm used by these tools has several hyperparameters that need to be tuned to achieve optimal performance. Different tools use different approaches to optimize these hyperparameters. For example, PyCaret uses a combination

of grid search and random search, while Auto-Sklearn 2.0 uses Bayesian optimization. The effectiveness of these methods in finding optimal hyperparameters can vary.

Time spent training: The amount of time spent training can also affect the RMSE score. With more time, the models can potentially be trained to a higher level of accuracy. As we can see from the metrics, Auto-Sklearn 2.0 and AutoGluon achieved a decrease in RMSE values when given more time to train.

Dataset characteristics: The performance of ML models is heavily influenced by the characteristics of the dataset, such as the size, complexity, and distribution of the data. Different algorithms and hyperparameters may perform better or worse on different types of data.

In summary, the differences in RMSE/MAPE scores among the tools studied can be traced back to differences in algorithms, hyperparameter optimization methods, training time, and specific traits of the dataset. A close look at the RMSE scores related to the Bitcoin dataset shows that PyCaret achieves the lowest value of 10,311, followed by Auto-Sklearn 2.0 and AutoGluon. This highlights PyCaret's ability to provide the most accurate forecasts according to the chosen metric, a finding that is also reflected in the MAPE values.

It is important to note that the RMSE/MAPE scores from Auto-Sklearn 2.0 and AutoGluon are also relatively moderate, especially when longer training times are employed. This supports the idea that these tools can also be effective in modeling the Bitcoin dataset, although other factors like ease of use and computational resources should be considered when choosing a tool.

Given the volatile nature of the Bitcoin dataset, it is reasonable to assume that model accuracy may decline over time due to market changes. Therefore, ongoing evaluation and improvement are essential to maintain the reliability and usefulness of these models, ensuring their long-term effectiveness.

The weather dataset (Delhi):

PyCaret uses a range of ML algorithms to automatically train and evaluate models on a given dataset. In this case, PyCaret achieved an RMSE score of 3.10 and a MAPE score of 0.10 using the extra trees regressor algorithm. The extra trees algorithm is a type of decision tree that uses randomization to improve the accuracy and stability of the model. The specific hyperparameters chosen by PyCaret, as well as the size and complexity of the dataset, may have contributed to the specific RMSE score achieved.

Auto-Sklearn 2.0 is a tool that uses Bayesian optimization to select the best combination of ML algorithms and hyperparameters for a given dataset. In this case, Auto-Sklearn 2.0 achieved the lowest RMSE score of 2.07 after 3600 s of training and achieved the lowest MAPE score of 0.07 using the gradient boosting algorithm. Gradient boosting is an ensemble learning technique that combines multiple weak models to create a more accurate overall model. The use of Bayesian optimization allows Auto-Sklearn 2.0 to efficiently explore the space of possible models and hyperparameters to find the best combination for the given dataset.

AutoGluon achieved low RMSE/MAPE scores across a range of quality levels, with the lowest score achieved at the highest quality level using an ensemble model. Like gradient boosting, ensemble models combine multiple models to create a more accurate overall model. The specific hyperparameters chosen by AutoGluon, as well as the size and complexity of the dataset, may have contributed to the specific RMSE/MAPE scores achieved.

Overall, Auto-Sklearn 2.0 achieved the lowest RMSE/MAPE score and appears to have found the most accurate model for the weather in Delhi dataset. AutoGluon also achieved relatively low scores across a range of quality levels, while PyCaret achieved a reasonable score but was not as accurate as the other tools. However, it is worth noting that the specific hyperparameters chosen by each tool, as well as the size and complexity of the dataset, likely contribute to the specific RMSE/MAPE scores achieved, and other metrics such as accuracy and precision should also be considered when selecting an ML model.

COVID-19 Dataset:

PyCaret achieved a low RMSE/MAPE score using the orthogonal matching pursuit algorithm for predicting COVID-19 deaths.

Auto-Sklearn 2.0 struggled to find an accurate model for the high complexity of COVID-19 data, and a large number of variables might have impacted the prediction of the number of deaths.

AutoGluon used an ensemble approach, combining the predictions of multiple models to achieve higher accuracy. Its low RMSE/MAPE scores across a range of quality levels suggest that it is well-suited for predicting COVID-19 deaths.

Overall, PyCaret achieved the lowest RMSE score and appears to have found the most accurate model for the COVID-19 dataset. AutoGluon obtained relatively low scores across various quality levels in root mean square error (RMSE), and it exhibited the lowest Mean Absolute Percentage Error (MAPE) score when trained with the 'best quality', while Auto-Sklearn 2.0 achieved higher scores and was not as accurate for this dataset. While comparing the RMSE/MAPE scores between these tools can provide some insight into their performance, it is important to keep in mind that the hyperparameters chosen and the size and complexity of the dataset can have a significant impact on the results. Therefore, it may be necessary to consider other metrics, such as accuracy and precision, in addition to RMSE/MAPE, when selecting an ML model for a particular task.

Limitations of the Study:

One limitation we faced was the highly specialized and rapidly evolving nature of the field. The field of AutoML is advancing rapidly, with new techniques, algorithms, and tools being developed at a fast pace. This dense and dynamic nature of the field presented challenges in terms of keeping up with the latest advancements, understanding the nuances of different AutoML techniques, and thoroughly evaluating the performance of various AutoML tools.

While metrics are commonly used to evaluate model performance, they may not always accurately identify a good model, as they may not fully capture the complexities of real-world scenarios, may prioritize different aspects of performance, and may not align with project requirements or domain-specific considerations. It is important to interpret the results of performance metrics with caution and consider other relevant factors when evaluating the quality of AutoML models.

Another limitation of our study was the significant impact of the datasets used for training and testing the AutoML. The size and characteristics of the datasets can greatly affect the performance and accuracy of the AutoML models. When using large datasets, training the models can be time-consuming and resource-intensive. This can sometimes result in having to stop the training prematurely without obtaining meaningful results. Additionally, large datasets may pose challenges in terms of formatting and handling and may require additional preprocessing steps to make them compatible with the AutoML tools.

On the other hand, small datasets, such as the COVID-19 dataset we used, can also present challenges. With limited data points, the AutoML models may struggle to make accurate predictions, as they may not have enough examples from which they can learn. Additionally, the nature of the data, such as the exponential growth in COVID-19 deaths, can also pose challenges for the models to capture the patterns effectively.

Furthermore, some datasets, such as the Bitcoin dataset, may be inherently volatile and unpredictable, making it difficult for AutoML to generate meaningful predictions. In such cases, the limitations of the dataset itself can hinder the performance of the models generated by the AutoML tool or any ML model that was created manually.

This paper evaluates multiple AutoML tools regarding how well they perform in analyzing time series datasets. The chosen tools were tested on metrics explained previously in the methodology, but there are other metrics that need examination as well [39]. These other metrics are not the scores given as to how accurately they are predicted. Instead, they are how polished and well-made the tools are for every kind of user. Not only the tool itself

but also the documentation and ease of use are to be evaluated since that is something for which AutoML is made. AutoML is made to alleviate the burden of knowing how to create and train an ML model, something that the average user/customer would not know how to accomplish. New studies started to emerge providing open-source code for the quick adaptation and comparison of AutoML tools [40].

In our research, we operated under the assumption that the developers behind each AutoML tool had already optimized the hyperparameters to enhance performance. Therefore, we refrained from tweaking these presets, positing that they were calibrated for peak efficiency. Our assessments were carried out with each tool in its default state, mirroring the common use case for individuals with limited ML expertise. This approach provided us with valuable insights into how each tool performs out of the box, which is especially relevant for users unfamiliar with the intricacies of ML fine-tuning. Considering the vast customization capabilities of AutoML tools, attempting to standardize their settings across different algorithms would be unnecessarily complicated and typically unfeasible.

6. Conclusions

This study conducts a comprehensive review of various AutoML tools in relation to time series datasets, aiming to assess their effectiveness for time series analysis. The research method involves a detailed comparison of three specific AutoML tools across different datasets using a set of strict evaluation metrics. The results have the potential to guide organizations that work with time series data in choosing the most appropriate AutoML tools for their specific needs.

This research confirms the usefulness of AutoML tools in the field of time series analysis. However, it is essential to recognize that performance varies depending on both the AutoML tool used and the particular dataset analyzed. This performance variability highlights the complex factors at play. Some AutoML tools may have specific requirements for dataset suitability, while others may not be as user friendly as advertised.

Regarding the current state of AutoML tools, this study suggests there is room for improvement, particularly in user-friendliness and the expansion of evaluation metrics. The research also anticipates future work that will validate a broader range of AutoML tools, providing insights beyond the scope of this current study. Future research is expected to include a larger number of datasets, extended timelines for experimentation, and additional evaluation metrics.

In summary, this study serves as a valuable resource by offering in-depth insights into the practical application of AutoML tools in the context of time series analysis. It underscores the continual necessity for research and development within this specialized field. The research includes a meticulous analysis of three distinct datasets pertinent to time series analysis, as well as an overview of relevant evaluation metrics. We anticipate that this research will serve as a guiding framework for future academic studies and offer a practical direction to organizations looking to employ AutoML tools for time series analysis and forecasting.

Author Contributions: G.W., U.E., O.A.M., and S.M.L.: software, formal analysis, writing—original draft, visualization. O.T.: conceptualization, methodology, resources, writing—review and editing, supervision. T.C.A.: conceptualization, methodology, resources, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, X.; Liao, H.; Tang, M. Decision making towards large-scale alternatives from multiple online platforms by a multivariate time-series-based method. *Expert Syst. Appl.* **2022**, *212*, 118838. [CrossRef]
2. Li, L.; Yan, J.; Zhang, Y.; Zhang, J.; Bao, J.; Jin, Y.; Yang, X. Learning Generative RNN-ODE for Collaborative Time-Series and Event Sequence Forecasting. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 7118–7137. [CrossRef]
3. Ensafi, Y.; Amin, S.H.; Zhang, G.; Shah, B. Time-series forecasting of seasonal items sales using machine learning—A comparative analysis. *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100058. [CrossRef]
4. Yu, Y.; Zeng, X.; Xue, X.; Ma, J. LSTM-Based Intrusion Detection System for VANETs: A Time Series Classification Approach to False Message Detection. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23906–23918. [CrossRef]
5. Sagana, C.; Devi, M.; Sangeetha, M.; Shwetha, K.; Devi, M.S.Y.; Udhayanidhi, C. Smart Weather Forecasting using Machine Learning Approach. In Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 29–31 March 2022; pp. 1072–1078. [CrossRef]
6. Alghanmi, N.; Alotaibi, R.; Buhari, S.M. Machine Learning Approaches for Anomaly Detection in IoT: An Overview and Future Research Directions. *Wirel. Pers. Commun.* **2021**, *122*, 2309–2324. [CrossRef]
7. Tiwari, S.; Chanak, P.; Singh, S.K. A Review of the Machine Learning Algorithms for Covid-19 Case Analysis. *IEEE Trans. Artif. Intell.* **2023**, *4*, 44–59. [CrossRef]
8. Akinci, T.C.; Topsakal, O.; Wernerbach, A. Machine learning-based wind speed time series analysis. In Proceedings of the 2022 Global Energy Conference (GEC), Batman, Turkey, 26–29 October 2022; pp. 391–394. [CrossRef]
9. Gao, H.; Qiu, B.; Barroso, R.J.D.; Hussain, W.; Xu, Y.; Wang, X. TSMAE: A Novel Anomaly Detection Approach for Internet of Things Time Series Data Using Memory-Augmented Autoencoder. *IEEE Trans. Netw. Sci. Eng.* **2022**, *10*, 2978–2990. [CrossRef]
10. Bahri, M.Z.; Vahidnia, S. Time Series Forecasting Using Smoothing Ensemble Empirical Mode Decomposition and Machine Learning Techniques. In Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 16–18 November 2022; pp. 1–6. [CrossRef]
11. Doke, A.; Gaikwad, M. Survey on Automated Machine Learning (AutoML) and Meta learning. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–5. [CrossRef]
12. Alsharef, A.; Aggarwal, K.; Sonia; Kumar, M.; Mishra, A. Review of ML and AutoML Solutions to Forecast Time-Series Data. *Arch. Comput. Methods Eng.* **2022**, *29*, 5297–5311. [CrossRef]
13. Alsharef, A.; Sonia, S.; Kumar, K.; Iwendi, C. Time Series Data Modeling Using Advanced Machine Learning and AutoML. *Sustainability* **2022**, *14*, 15292. [CrossRef]
14. Mukherjee, S.; Rao, Y.S. Auto-ML Web-application for Automated Machine Learning Algorithm Training and evaluation. In Proceedings of the 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 7–9 April 2022; pp. 1–6. [CrossRef]
15. Cap, H.-A.; Do, T.-H.; Lakew, D.S.; Cho, S. Building a Time-Series Forecast Model with Automated Machine Learning for Heart Rate Forecasting Problem. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 1097–1100. [CrossRef]
16. Uras, N.; Marchesi, L.; Marchesi, M.; Tonelli, R. Forecasting Bitcoin closing price series using linear regression and neural networks models. *PeerJ Comput. Sci.* **2020**, *6*, e279. [CrossRef]
17. Hiransha, M.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Comput. Sci.* **2018**, *132*, 1351–1362. [CrossRef]
18. Hanussek, M.; Blohm, M.; Kintz, M. Can AutoML outperform humans? An evaluation on popular OpenML datasets using AutoML Benchmark. In Proceedings of the 2020 2nd International Conference on Artificial Intelligence, Robotics and Control, in AIRC'20, Online, 26–28 December 2020; Association for Computing Machinery: New York, NY, USA, 2021; pp. 29–32. [CrossRef]
19. Paldino, G.M.; De Stefani, J.; De Caro, F.; Bontempi, G. Does AutoML Outperform Naive Forecasting? *Eng. Proc.* **2021**, *5*, 36. [CrossRef]
20. Tealab, A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Futur. Comput. Informatics J.* **2018**, *3*, 334–340. [CrossRef]
21. Paladino, L.M.; Hughes, A.; Perera, A.; Topsakal, O.; Akinci, T.C. Evaluating the Performance of Automated Machine Learning (AutoML) Tools for Heart Disease Diagnosis and Prediction. *AI* **2023**, *4*, 1036–1058. [CrossRef]
22. Idrees, S.M.; Alam, M.A.; Agarwal, P. A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access* **2019**, *7*, 17287–17298. [CrossRef]
23. Javeri, I.Y.; Toutiaee, M.; Arpinar, I.B.; Miller, J.A.; Miller, T.W. Improving Neural Networks for Time-Series Forecasting using Data Augmentation and AutoML. In Proceedings of the 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 23–26 August 2021; pp. 1–8. [CrossRef]
24. Zhao, R.; Wang, J.; Chen, G.; Li, Q.; Yuan, Y. A Machine Learning Pipeline Generation Approach for Data Analysis. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1488–1493. [CrossRef]
25. Ono, J.P.; Castelo, S.; Lopez, R.; Bertini, E.; Freire, J.; Silva, C. PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 390–400. [CrossRef]

26. Nguyen, D.A.; Kononova, A.V.; Menzel, S.; Sendhoff, B.; Back, T. An Efficient Contesting Procedure for AutoML Optimization. *IEEE Access* **2022**, *10*, 75754–75771. [CrossRef]
27. Parmentier, L.; Nicol, O.; Jourdan, L.; Kessaci, M.-E. TPOT-SH: A Faster Optimization Algorithm to Solve the AutoML Problem on Large Datasets. In Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 4–6 November 2019; pp. 471–478. [CrossRef]
28. Sarafanov, M.; Pokrovskii, V.; Nikitin, N.O. Evolutionary Automated Machine Learning for Multi-Scale Decomposition and Forecasting of Sensor Time Series. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2022; pp. 01–08. [CrossRef]
29. Gawalska, A.; Czub, N.; Sapa, M.; Kołaczkowski, M.; Bucki, A.; Mendyk, A. Application of automated machine learning in the identification of multi-target-directed ligands blocking PDE4B, PDE8A, and TRPA1 with potential use in the treatment of asthma and COPD. *Mol. Inform.* **2023**, *42*, 2200214. [CrossRef]
30. Ruf, P.; Madan, M.; Reich, C.; Ould-Abdeslam, D. Demystifying MLOps and Presenting a Recipe for the Selection of Open-Source Tools. *Appl. Sci.* **2021**, *11*, 8861. [CrossRef]
31. Qi, W.; Xu, C.; Xu, X. AutoGluon: A revolutionary framework for landslide hazard analysis. *Nat. Hazards Res.* **2021**, *1*, 103–108. [CrossRef]
32. Sreyes, K.; Davis, D.; Jayapandian, N. Internet of Things and Cloud Computing Involvement Microsoft Azure Platform. In Proceedings of the 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 13–15 October 2022; pp. 603–609. [CrossRef]
33. Rastogi, R. Machine Learning @ Amazon. In Proceedings of the 2017 IEEE 24th International Conference on High-Performance Computing (HiPC), Jaipur, India, 18–21 December 2017; p. 182. [CrossRef]
34. Meisenbacher, S.; Turowski, M.; Phipps, K.; Rätz, M.; Müller, D.; Hagenmeyer, V.; Mikut, R. Review of automated time series forecasting pipelines. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2022**, *12*, e1475. [CrossRef]
35. Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. *arXiv* **2020**, arXiv:2003.06505. [CrossRef]
36. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum, M.; Hutter, F. Auto-sklearn: Efficient and Robust Automated Machine Learning. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 113–134. [CrossRef]
37. Pol, U.R.; Sawant, T.U. Automl: Building a classification model with PyCaret. *YMER* **2021**, *20*, 547–552.
38. Sarangpure, N.; Dhamde, V.; Roge, A.; Doye, J.; Patle, S.; Tamboli, S. Automating the Machine Learning Process using PyCaret and Streamlit. In Proceedings of the 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 3–5 March 2023; pp. 1–5. [CrossRef]
39. Lenkala, S.; Marry, R.; Gopovaram, S.R.; Akinci, T.C.; Topsakal, O. Comparison of Automated Machine Learning (AutoML) Tools for Epileptic Seizure Detection Using Electroencephalograms (EEG). *Computers* **2023**, *12*, 197. [CrossRef]
40. Topsakal, O.; Akinci, T.C. Classification and Regression Using Automatic Machine Learning (AutoML)—Open Source Code for Quick Adaptation and Comparison. *Balk. J. Electr. Comput. Eng.* **2023**, *11*, 257–261. [CrossRef]