Journal of Statistics & Applied Science, Issue - 6
ISSN 2718-0999

Research Article

Received: date:27.10.2022
Accepted: date:15.12.2022
Published: date:31.12.2022

Comparison of Time Series Models for Predicting Online Gaming Company Revenue

Utku Can Aytaç [1*], Tayfun Küçükyılmaz[2] and Gönenç Seçil Tarakçıoğlu[3]

[1] Computer Engineering Department, Istanbul Aydın University, Istanbul, Turkey; utkuaytac@stu.aydin.edu.tr
[2] Business Information Management, Erasmus University, Rotterdam, Netherlands; kucukyilmaz@rsm.nl
[3]Triodor Ar-Ge Co. Ltd., Maltepe, Istanbul, Turkey; g.tarakcioglu@triodor.eu
Orcid: 0000-0002-0470-6460[1] Orcid: 0000-0002-2551-4740[2] Orcid: 0000-0002-6528-9496[3]
*Correspondence: utkuaytac@stu.aydin.edu.tr

Abstract: Online gaming industry is where the effects of any change can be seen in a concise time. Therefore, real-time analysis of revenues, analysis of the commercial performance of the developed content, and rapid monitoring of the revenue contributions of the improvements are essential. Therefore, financial forecasting is a crucial part of business plan, which can help strategize how much and how quickly the company intends to grow. In financial forecasting of a given time series, revenue estimations for the future will become important research in the industry. This research offers a detailed analysis of recent time series models and focused on both deep learning and statistical methods for time series forecasting on real-world revenue data. The results of the study are examined using one of the leading Finland-based online gaming companies' revenues data. In our experiments, we investigated various time series forecast techniques, such as SARIMA, Theta, Holt-Winters, Prophet, Dense Neural Network (DNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), N-Beats and Ensemble models. The experimental evaluations illustrate that deep learning models can optimize financial forecast operations. The result of the study provides insights to managers and analysts in determining the best model to adopt.

Keywords: Revenue forecast, time series analysis, SARIMA, Theta, Holt-Winters, Prophet; DNN, CNN; LSTM

1. Introduction

Throughout the last few decades, the use of time-series forecasting has spread across numerous domains, such as the stock market, power generations, and healthcare, amongst many others. With the rapid development and adaptation of technology, multiple methods have been introduced to perform task more accurately. Researchers are continuously attempting to find effective methods to extract significant information and utilize the extracted information in order to solve large real-life problems. To this date, numerous traditional statistical methods were applied for revenue forecasting. Autoregressive Intergrade Moving Average Model (ARIMA), Holt-Winters (HW) and Exponential Smoothing (ES) are some of the algorithms used by researchers with substantial success. In [1], these 3 methods were applied to the problem of oil-price forecasting problem. The study examined prices of oil for West Texas Intermediate crude. In the study, ARIMA performed the best in terms of different evaluation metrics. In a similar study, ARIMA has been successfully used to model the Kenyan GDP [2]. The results confirm the effectiveness of the technique, with a relative error of less than %5. Other trend analysis techniques, such as HW, have been used to forecast the revenue of the Bangabandhu

Citation: U. C. Aytaç , T. Kucukyilmaz ve G. S. Tarakcıoğlu , "Comparison of Time Series Models for Predicting Online Gaming Company Revenue", Journal of Statistics and Applied Sciences, no. 6, pp. 25-36, Dec. 2022, doi:10.52693/jsas.1195048

Multipurpose Bridge in Bangladesh [3]. The performance was evaluated using different statistical measures, and the study reported an average error of less than [4].
Due to computational constraints, many of the statistical methods could not be used effectively for more complex and high-dimensional real-world problems. In that respect, using more scalable techniques such as time series analysis or machine learning present a viable alternative for the revenue forecast problem. Trend analysis involves looking at data collected at different periods of time in order to identify and estimate changes through time. There are also studies in the literature that use machine learning techniques in trend analysis. Studies in the literature that uses machine learning for the revenue forecast problem can be further categorized into two with respect to their objectives: to perform forecasts or to infer content-based similarities for new products. For the former objective, past observations are used as features representing a state of the company/industry. For the latter objective, similar products and the level of similarity to these products can be inferred by using the past observations with the hope to act as a proxy on the product revenue.
Application of Machine Learning (ML) enabled various models which has led to significant advances for the revenue forecast problem. Recently, Deep Learning (DL) models were also employed to the problem, presenting results that significantly outperform their traditional ML counterparts. such studies were succeeded by deep learning methods which proved to generate much more accurate results by tuning hyper-parameters and choosing most suitable framework [4]. One of the deep learning-based techniques that attracted a lot of attention in the recent years is the Recurrent Neural Networks (RNN), and one of its variations, Long Short-Term Memory (LSTM). In [5], the authors established that LSTM outperforms the performance of ARIMA based models on 33 years historical monthly financial timeseries data from the Yahoo finance Website. Results show that LSTM-based models outperform ARIMA-based models with a high margin (i.e., between %84 -%87 reduction in error rates). In another study, LSTM used for the forecasting prices of Bitcoin and the results indicate significant performance gains [6] however, using RNN's are generally computationally expensive due to their sequential nature. To resolve these issues, convolutional neural networks (CNN) are evaluated as an alternative. In [7], a one-dimensional (1D) CNN was used for the prediction of a pharmaceutical company's sales using their real large-scale sales data. The results are evaluated using the Mean Absolute Error (MAE) and present accurate results, with an error of approximately %3.5. In conjunction to the above deep learning methods, Facebook recently introduced an open-source library known as Facebook Prophet in 2017 [8]. In another study [9], Facebook Prophet was used in the domain of retail business by classifying the product portfolio according to the expected level of forecasting reliability, it generated accurate monthly and quarterly sales forecasts; approximately %40 of the product portfolio which can be forecasted with MAPE < %15 on a quarterly basis. Facebook Prophet has also outperformed ARIMA in the forecasting of Bitcoin prices on a real-life dataset collected during 2012 and 2018 [10]. Another interesting application of Facebook Prophet was in the realm of agriculture [11] where, it was used to predict the sales of agricultural products to stabilize demand and supply. The performance of dense convolutional neural networks has also been critically analyzed over the years. In a study [12], CNN-based model was used for the prediction of sales for a pharmaceutical company. The company's real large-scale sales data was used, and a dense model was proposed. The model was evaluated based on mean absolute percent error (MAPE) and mean absolute error (MAE). In the study [13], CNN was proposed for the task of stock prediction. The proposed model consisted of 6 layers, including 2 fully connected layers, and the authors used rectified linear unit (ReLU) as well as leaky rectified linear unit (LReLU) as activation functions. It was evaluated on multiple stock datasets which allowed it to be concluded that it was a robust approach to the problem. In recent years, a pure deep learning method was introduced known as N-Beats [14]. It gained massive popularity after it outperformed as the winner of the competition M4 (Makridakis Competitions are open competitions to evaluate the accuracy of different time series forecasting algorithms). It was critically analyzed on multiple datasets [15], specifically on M4 which included data regarding finance. It was also used with the M5 dataset (unit sales of the Walmart by revenue), as a basis in a competition, which mainly consisted of retail sales [16]. The results reveal that the proposed techniques outperformed many other

models for the forecast problem and achieved second place in the competition. The use of hybrid and ensemble models is also a common practice to improve the performance of forecast techniques since the ensemble of individual models usually performs better than a single model. Several studies have utilized such models for a variety of tasks. In [17], an EEMD-based EELM ensemble learning algorithm was used to forecast prices of crude oil. It exhibited a higher level of prediction accuracy when compared to individual models. In [18], an ensemble of N-Beats and RNN was introduced known as N-Beats-RNN. This was analyzed on the M4 dataset and produce similar results to the N-Beats models. The major contribution of this study lays in the reduction of the training time which was reduced by almost 9 times with the use of the ensemble model. The motivation of performing trend analysis in this study is forecasting future revenue expectations of a product and presenting it as a report to the product owner with upper and lower forecast limits for potential revenues. To do so, we evaluated well-known statistical techniques such as SARIMA, Theta, Holt Winters, Prophet, and deep learning such as Dense Neural Network (DNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), N-Beats and Ensemble) models that can scale with large quantities of real life data, which would be a preferable choice for the problem for trend analysis: The future expectations would be established using machine learning models. The layout of this paper is as follows: The experimental settings, fine tuning, dataset preprocessing and introduced methods are given in Section 2. Section 3 describes the experimental results and compares the performance of the introduced methods and Section 4 describes discussion and future research directions.

2. Materials and Methods

In this section, we outline the methodology used in this work. We begin with general background knowledge of the field which experiments will be held. After giving additional statistical information of the data, preprocessing of the datasets are described. Each model goes through a specific training pipeline with the metrics used to compare the various modeling techniques employed.

2.1. Dataset and Processing

In the astonishing gaming industry with a global market value of more than 100 billion USD (larger than movie and music industries together), 2.2 billion gamers and hundreds of thousands of games released every year [19], the biggest problem companies face are reaching high volumes of users, in other words, revenue. Many game developers and publishers are focused on how to maximize the long-term engagement of players to increase revenue. Online gaming ecosystem creates many revenue channels for the industry and revenue channels are very complex and diverse. For example, an online game can be monetized through in-app purchases or advertisements, or it can be downloaded from the mobile store for a fee.

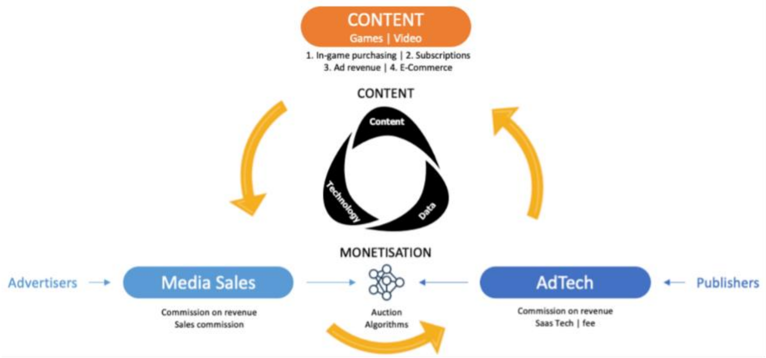


Figure 1. Revenue Ecosystem of Online Gaming.

Figure 1 illustrates the monetization ecosystem of online gaming. The market includes subscriptions, in-game content, online advertisement, and other revenue sources beyond the original price of the game itself. Especially, online advertisement (AdTech) is an important revenue source for publishers in general and specifically for the creative industries. Since AD exchange markets are highly dynamic and uncertain, the selection of an appropriate source is critical for small and medium enterprises, for which over %80 of the total revenue is collected from online advertising [20]. In the market, revenue forecast is critical for businesses because it allows you to plan how soon and how much you want to grow. Besides, if companies can forecast their revenues, publishers will be able to set proper budgets before running operations.

In our experiments, we used historical daily financial timeseries from January 2020 to October 2021 from one of the biggest online gaming companies in Finland. Figure 2 illustrates the real-world revenue forecasting benchmark data obtained for the company. IOS and Android in-app purchase revenues are shown in Figure 3 and 4, respectively. These figures clearly show that revenue expectations may reduced over time.

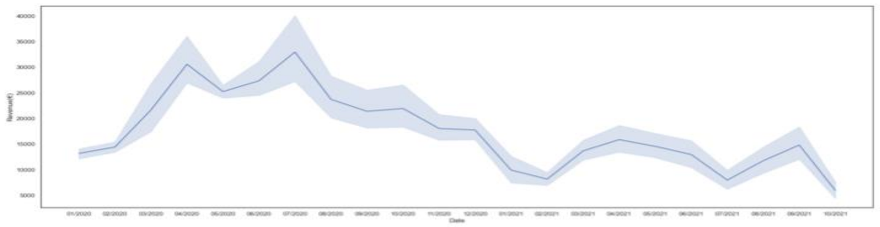


Figure 2. Revenue between 2020 January-2021 October.

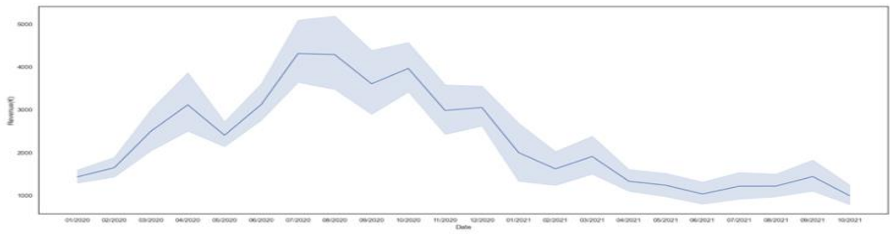


Figure 3. Android IAP revenue between 2020 January-2021 October.

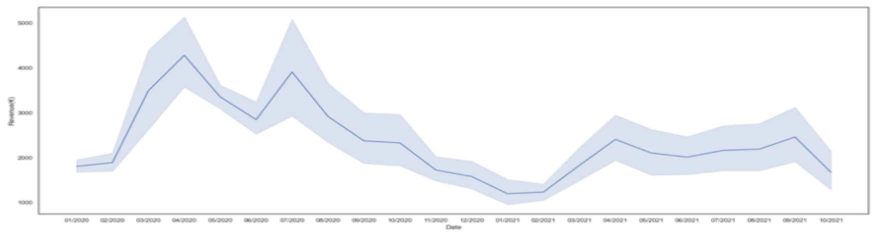
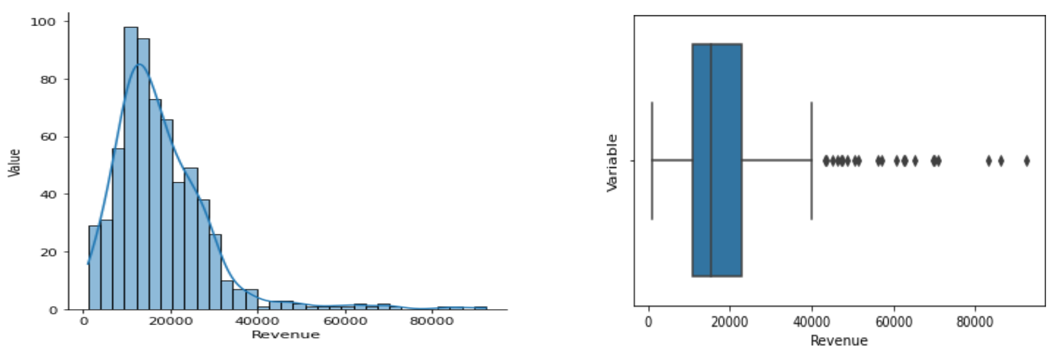


Figure 4. IOS IAP revenue between 2020 January-2021 October.

While we go through the data, further statistical data analysis was needed before the experiments to decide which models would be used. First, we investigated the data distribution, because most machine learning models are designed to work best with particular types of data distributions. As Figure 5 shows, real life distributions are usually skewed. The peak of the histogram and box plot veers to the left, and the dataset can be described as being positively skewed to the right.



a) b)

Figure 5. Graphical representations for the frequency of numeric data values: (**a**) The number of data points that fall within a specified range of values. (**b**) The skewness of numerical data and its distribution can be seen by showing the data quartiles (or percentiles) and averages.

The value distribution of a data set can be problematic to model because the tail region may represent an outlier of statistical models. Machine learning models are robust to outliers, but if the data are characterized only by statistical analysis, high-revenue situations might not be detected correctly by the distribution.

A time series is a collection of data points that have been indexed in time. Because a time series of data is ordered, it frequently exhibits serial dependence. When the value of one data point at one moment is statistically dependent on the value of another data point at another time, this is known as serial dependence. This property of time series data contradicts one of many statistical studies' essential assumptions, namely that of statistical independence of observations. For the sake of comparison, we investigated the similarity between independent variables in our data set. Figure 6 illustrates the autocorrelation with a %95 confidence interval. The x-axis displays the number of lags, and the y-axis displays the autocorrelation at that number of lags. By default, the plot starts at lag = 0, and the autocorrelation will always be 1 at lag = 0. The Pearson correlation coefficient has a value between -1 and 1, where 0 is no linear correlation, > 0 is a positive correlation, and < 0 is a negative correlation. The figure shows that revenue values are within the %95 confidence interval (represented by the solid gray line) for lags > 0, which confirms that there is no autocorrelation in that area.

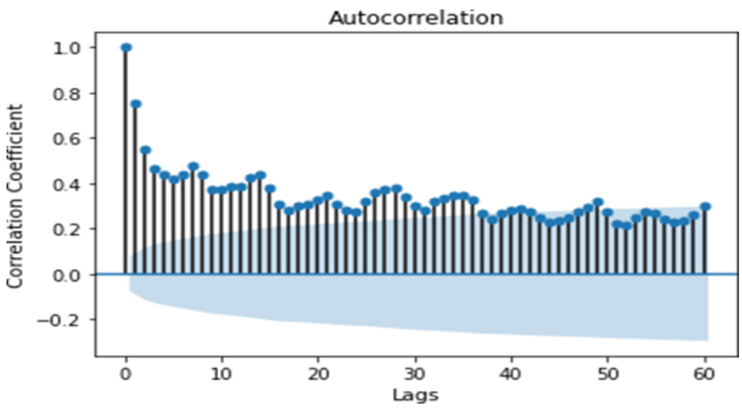


Figure 6. Correlation chart with %95 confidence interval.

Because the dataset used in experiments is univariate, rather than splitting the data set randomly into training and testing subsets as is traditional in machine learning exercises, the time series data were split sequentially. As shown in Figure 7 we created %80 training data subset and %20 test data subset. For the implementation, Tensorflow [21] and Keras [22] were used as deep learning application programming interfaces (APIs). The data preparation and visualization of the results were done using the Scikit-learn [23], NumPy [24], Matplotlib [25], and Seaborn [26] libraries. The simulations were computed on Google Colab [27], using an NVIDIA Tesla P100 GPU-based computing processor.

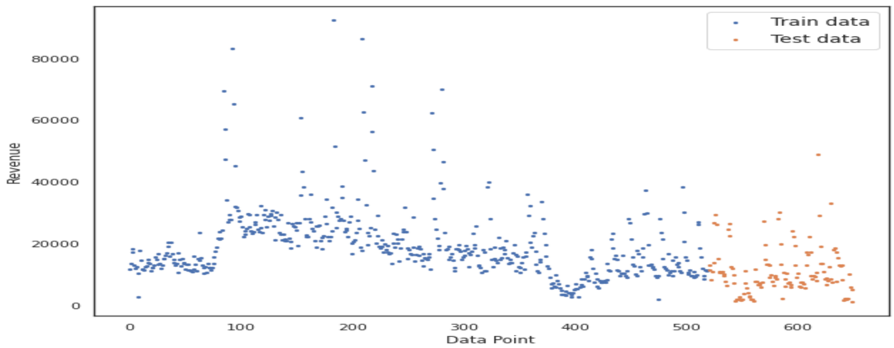


Figure 7. Train and test split for evaluating the performance of models.

3. Results

This section compares the prediction accuracies of the statistical and deep learning models with different parameters and time windows and provides robust complementary results of the proposed architectures with given specific performance metrics.

3.1. Performance Metrics

By using the current and past period values of the variable to be estimated by time series analysis, a forecasting model is obtained by various methods, and the future period values of the variable are estimated after the validity of this model is investigated. During our experiments, we will measure mean absolute error (MAE), root mean squared error (RMSE), mean absolute scaled error (MASE) and normalized root mean squared error (N-RMSE) to see the performance of the models.

MAE is the sum of the absolute values of the differences between all expected and predicted values divided by the total number of predictions n [28].

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \tag{1}$$

RMSE is defined as the square root of the mean square of the errors. The main difference between MAE and RMSE is, due to the squaring operation in RMSE, very small values (between 0 and 1) become even smaller, and larger values become even larger [29].

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_i - \hat{y}_j)^2} \tag{2}$$

This means that large error values will be expanded, and small error values will be ignored. RMSE is used when small errors can be safely ignored, and large errors should be penalized and reduced as much as possible. This is a useful metric if outliers more important. RMSE attaches great importance to large errors, and the model tries to minimize them as much as possible. MASE is a scale-free error metric that gives each error as a ratio compared to a baseline's average error [30]. This metric is used for time series predictions to determine the effectiveness of the models with the output of naïve forecasting approach which is generated by current forecast to the output of last time step.

$$\text{MASE} = \text{mean}\ (|q_t|) \tag{3}$$

Where: $$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \tag{4}$$

3.1. Experiment Results

A time series can be defined as a series of measurements observed at periodic time intervals, and a large amount of reliable data is needed to obtain good results in analysis methods. Therefore, it is very important to provide the necessary assumptions for these methods. While performing time series analysis, the time-dependent changes of the data must be examined. In time series forecast experiments for deep learning, we used the timesteps of 7,30,60 days respectively; from the past into the future, taking Horizon as day 1. In the time series literature, this is also called the windowing method (windowing for one week (window=7) to predict the next single value (horizon=1). It is a method to turn a time series dataset into supervised learning. The comparative analysis is conducted by three performance indicators: RMSE, MAE and MASE are used to compare the forecasting performance of the models and their components. In the first step, the overall performance of the models is compared. Table 1 illustrates the network architecture of the models.

Table 1. Network architecture and hyperparameter of deep learning models.

DL Models	Architecture with Parameters
CNN	• First layer: filters=128, kernel size=7, Padding= Causal, Activation= Relu • Second layer: filters=256, kernel size=5, Padding= Causal, activation= Relu • Batch size=128 • Epochs=100
DNN	• First layer: filters=128, Activation= Relu • Second layer: filters=256, Activation= Relu • Batch size=128 • Epochs=100
LSTM	• First LSTM layer: filters=128, Activation= Relu • Second Dense layer:filters=256, Activation= Relu • Batch size=128 • Epochs=100
Ensemble	• First layer: filters=128, kernel initializer= he normal, Activation= Relu • Second layer: filters=128, kernel initial- izer= he normal, Activation= Relu • Ensemble loss functions= MSE, MAPE
N-Beats	• N neurons= 128,n layers = 4 • N Neurons = 512 • N Layers = 4 • N Stacks = 30 • Batch size=1024 • Epochs=5000

Table 2, Table 3 and Table 4 present the overall performance of the deep learning models with 7, 30 and 60 days window respectively. It can be observed that all the models which determines the effectiveness of forecasts generated through an algorithm performed well (MASE < 1) by comparing the predictions with the output of a naive forecasting approach. As we know that RMSE punishes large errors more than MAE, so if we want to discard outlier potential data, we can check the MAE results, which Ensemble performed well.

Table 2. Accuracy analysis of the deep learning models with 7 days window.

DL Models	RMSE	MAE	MASE
DNN	8965.51	4885.56	0.80
CNN	8939.43	4858.05	0.80
LSTM	8745.04	4726.48	0.78
Ensemble	8806.87	4698.88	0.77
N-Beats	8814.02	4852.48	0.80

Obviously, the larger the window, the more information you can consider about the time series. However, it can reduce the sensitivity of the system, leading to smoother predictions. In our opinion, the optimal window size corresponds to the dimension of the fractal data. As it can be seen in Table 3 and Table 4, expanding the window size incremented the MASE values which is ineffective. The testing results are satisfactory for the experiment that window size of 7 can be used for all the deep learning models. Deep learning model analysis indicates that Ensemble, LSTM and N-Beats algorithm performed well.

Table 3. Accuracy analysis of the deep learning models with 30 days window.

DL Models	RMSE	MAE	MASE
DNN	5917.67	4144.67	0.91
CNN	5894.00	3876.74	0.85
LSTM	5773.62	3959.81	0.87
Ensemble	7089.28	3952.95	0.84
N-Beats	7373.83	4403.14	0.93

Table 4. Accuracy analysis of the deep learning models with 60 days window.

DL Models	RMSE	MAE	MASE
DNN	5989.84	4218.74	0.96
CNN	6211.45	4452.02	1.01
LSTM	5974.56	4387.61	1.00
Ensemble	6152.99	3809.62	0.98
N-Beats	6945.09	4466.54	1.15

In order to compare the statistical forecasting methods, we fit each of the time-series forecasting methods with different parameters shown in Table 5, using the training data and measure the fitted model on the testing data to determine each methods forecasting accuracy.

Table 5. Statistical model parameters explanation.

Statistical Models	Parameters
SARIMA	p: Integer for trend autoregressive (AR) order. Q: Integer for trend difference order. D: Integer for trend moving average (MA) order. Trend: Deterministic trend. Can be ‘c’ (constant), ‘t’ (linear trend with time), ‘ct’ (both constant and linear trend)
Theta	M: Number of observations before the seasonal pattern repeats
Holt-Winters	Trend: Specifies the type of trend component. Can be ’add’ and ’mul’ or ’additive’ and ’multiplicative’ Seasonal: Specifies the type of seasonal component Can be ’add’ and ’mul’ or ’additive’ and ’multiplicative’ P: Seasonal periods
Prophet	S: Specifies the type of trend component. Can be ’add’ and ’mul’ or ’additive’ and ’multiplicative’

Table 6 compares the performance and accuracy of the statistical methods with different parameters. The following can be observed:

- ❖ SARIMA performs better in terms of RMSE.
- ❖ Next best performing model is the Prophet method, which may also be considered for forecasting revenues.
- ❖ None of the statistical methods performed well then deep learning models.

Table 6. Accuracy analysis of the statistical models with different parameters.

Statistical Models	Parameters	RMSE	MAE	MASE
SARIMA	p=2,d=1,q=1,trend=’ct’	7998.47	6382.33 14642.69 5331.67	1.42
	p=1,d=2,q=2,trend=’c’	18332.25		3.25
	p=2,d=1,q=1,trend=’t’	7895.50		1.18
Theta	m=12	58056.25	5643.02 5882.51 11067.58	1.25
	m=6	8388.31		1.31
	m=9	13036.27		2.46
Holt-Winters	Trend=’add’,s1=’mul’,p=7	33368.88	6382.33 28735.32 28735.32	1
	Trend=’mul’,s1=’add’,p=4	33368.88		6.38
	Trend=’add’,s1=’add’,p=7	33368.88		6.3

[1]s=seasonal

We summarize Table 7 by showing the best-performing model of time-series models. We can observe that the LSTM method performs the best in RMSE, followed by Ensemble in MAE and MASE.

Table 7. Best performing 3 models

DL Models	RMSE	MAE	MASE
N-Beats	8814.02	4852.48	0.80
LSTM	8745.04	4726.48	0.78
Ensemble	8806.87	4698.88	0.77

The comparison reveals that the average forecasting error for MAE and MASE in the test datasets is lower in the ENSEMBLE and LSTM models. Expanding the window size did not show too much effect. For example, in the N-Beats model, in MAE and MASE terms, the forecasting results of the series models using the test dataset increased by 0.80 to 0.93. Tables 5 compare the performance and accuracy of the statistical methods with different parameters. The results show that applying different parameters to all four statistical models, on average, did not improve the forecasting accuracy. According to the analytical results, the accuracy of the deep learning models are better than those of the statistical models in overall performance. From the above comparative analyses, best performing models are seen in Table 5. The models can be ranked as: (1) Ensemble (2) LSTM and (3) N-Beats.

4. Discussion

This study evaluates different time-series forecasting methods: SARIMA, Theta, Holt Winters, Prophet, Dense Neural Network (DNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), N-Beats and Ensemble models on univariate time-series forecast data. One of the main goals of this study is to help revenue forecasters, finance analysts, and managers understand how to compare and choose methods for revenue forecasting. These methods are chosen according to their abilities to identify complex relationships among time series data with trends. In this paper, we present the results of an empirical comparative evaluation of the performance of time series analysis methods for short-term revenue forecasting applied to real-world aggregated data for a Finnish gaming company. Based on the evaluation metrics, we conclude that forecasting sales using deep learning methods yields the best performance, on average, in comparison to statistical methods. Our findings suggest that for statistical (data-driven) methods with external parameters, Prophet is the best method to use for time series forecasting. For deep learning (data-driven) methods with a given set of parameters, ensemble method is the best model to use for time series forecasting of revenues. This conclusion is based on the results of a comparison of the root mean squared error (MASE) values obtained. Computationally, the N-BEATS method take much longer to fit to a data set than a other methods—approximately 10 minutes for a small data set. Large data sets are likely to take much longer to fit. Because of their reliance on external parameters, SARIMA modeling require training to use and may not be easily accessible to a novice. These two shortcomings may make SARIMA modeling less appealing to decision makers who are not comfortable with statistics. A decision maker and an organization need to evaluate the trade-off between forecasting accuracy and the abovementioned shortcomings. We also show that the Holt–Winters method and SARIMA modeling are both incapable of fully capturing the behavior of revenue time series. The impact of other external factors on the reliability of time series forecasts needs to be examined further. To generalize the results from this study, the methods considered need to be tested further on time series with different types of attributes. From an advertising perspective, representing the capabilities and functionalities of the experiments described here to potential product owners is an important asset. However, representing the functionalities of the methods examined here for potential products would require collection of data for products that are not associated with the proposed framework.

5. Conclusions

Revenue channels in the online gaming industry are very complex and diverse. An online game can be monetized through in-app purchases or advertisements, or it can be downloaded from the mobile store for a fee. Revenues generated through purchases are processed and reported to companies by AppleStore, GoogleStore and other payment service providers. Advertising-based revenues are much more complex and flow through APIs of different ad provider systems. Data collection methods, frequencies, data structures and technologies are also different from each other. This makes it difficult to collect and display the data. On the other hand, the online gaming industry is an area where the effects of any change can be seen in a very short time. Real-time analysis of revenues, analysis of the commercial performance of the developed content, and rapid monitoring of the revenue contributions of the improvements are essential. For an online gaming company which has several games in platforms, it is necessary to make comparisons and analyzes to determine development costs and to detect and stop developing unpromising products in a timely manner. To solve this issue, we tried to analyze which models best fit the univariate time series problem and provided results.

In the future, we intend to develop approaches to making statistical inferences about product revenues that can be categorized into two groups: anomaly detection of product revenues and trend analysis of product revenues. An anomaly is defined as an unexpected or erroneous behavior on some data. Anomaly detection for future forecasts may also provide very important information for product owners; potentially allowing them to manipulate the maintenance of their products in ways that can maximize the revenue obtained from each product. The proposed framework can collect information on registered products and their revenues on a continuous real-time basis. Important observations that can be reported to product owners are unexpected changes in the potential revenues of their products. Therefore, we propose integrating anomaly detection techniques into the proposed framework. First, the literature on anomaly detection will be reviewed, and several classification techniques will be proposed for anomaly detection. The techniques will be integrated into the framework and presented as potential means of reporting anomalies to product owners.

We will also elaborate on similarity-based multi-product trend analysis, which is possible after establishing the similarity between a given product and other products in the system. The effectiveness of similarity estimation and performance of the models will be evaluated with the objective of achieving the highest possible accuracy in identifying future product revenues. Additionally, the granularity of the data used to establish similarities will be examined so that the framework can achieve acceptable performance while respecting the privacy of all product owners. During the development, representational models for both future products and product owners will be developed. We intend for these models to contain features representing the potential player base of the products, their use patterns, and their interest patterns. The results of such analyses can be valuable for any product owner. We hope to identify which of these attributes would be valued assets of product owners and integrate these assets into the framework of statistical reports provided to the product owners.

Author Contributions: Conceptualization, T.K., GST and U.C.A.; methodology, T.K.; software, U.C.A.; validation, G.S.T, T.K. and U.C.A; formal analysis, T.K.; investigation, G.S.T.; resources, U.C.A; data curation, T.K.; writing—original draft preparation, U.C.A.; writing—review and editing, G.S.T..; visualization, U.C.A.; supervision, T.K.; project administration, G.S.T.; funding acquisition, U.C.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Scientific and Technological Research Council of Turkey (TUBITAK) 1501, grant number 3211378.

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

References

[1] G. A. Tularam, T. Saeed, "Oil-price forecasting based on various univariate time-series models" American Journal of Operations Research, 6 (03), 2016, 226.

[2] M. S. Wabomba, M. P. Mutwiri and F. Mungai, "Modeling and forecasting Kenyan GDP using autoregressive integrated moving average (ARIMA) models" Science Journal of Applied Mathematics and Statistics, 4(2), 2016, pp 64-73.

[3] M. H. Rahman, U. Salma, M. M. Hossain and M. T. F. Khan, "Revenue forecasting using holt–winters exponential smoothing", Research Reviews: Journal of Statistics, 5(3), 2016, pp 19-25.

[4] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez and A. Troncoso, "Deep Learning for Time Series Forecasting: A Survey", Big Data, 9 (1), 2021, pp 3-21.

[5] S. Siami-Namini, N. Tavakoli and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series", In 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018 pp. 1394-1401.

[6] C. H. Wu, C. C. Lu, Y. F. Ma and R. S. Lu, "A New Forecasting Framework for Bitcoin Price with LSTM", IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 168-175.

[7] V. Buyar, A. Abdel-Raouf, " A Convolutional Neural Networks-based Model for Sales Prediction", In Proceedings of the International Conference on Artificial Intelligence, Robotics and Control 2019, pp. 61-67.

[8] S.J. Taylor and B. Letham, "Forecasting at scale", The American Statistician, 72(1), 2018, pp 37-45.

[9] E. Zunic, K. Korjenic, K. Hodzic and D. Donko, "Application of facebook's prophet algorithm for successful sales forecasting based on real-world data" 2018.

[10] I. Yenidoğan, A. Çayir, O. Kozan, T. Dağ and Ç. Arslan, "Bitcoin forecasting using ARIMA and prophet", In 2018 3rd International Conference on Computer Science and Engineering, (UBMK), 2018, pp. 621-624.

[11] T. W. Yoo and I. S. Oh, "Time Series Forecasting of Agricultural Products' Sales Volumes Based on Seasonal Long Short-Term Memory", Applied Sciences, 10 (22), 2020, 8169.

[12] L. Qian, Y. Fu, and T. Liu, "An Efficient Model Compression Method for CNN Based Object Detection," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 766-769.

[13] S. Chen, and H. He, "Stock prediction using convolutional neural network", In IOP Conference series: materials science and engineering 435 (1), IOP Publishing, 2018, p. 012026.

[14] B. N. Oreshkin, D. Carpov, N. Chapados and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting", arXiv preprint arXiv:1905.10437, 2018.

[15] B. N. Oreshkin, D. Carpov, N. Chapados and Y. Bengio, "Meta-learning framework with applications to zero-shot time-series forecasting", arXiv preprint arXiv:2002.02887, 2020.

[16] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M5 accuracy competition: Results, findings and conclusions", Int J Forecast, 2020.

[17] V. Buyar and A. Abdel-Raouf, " A Convolutional Neural Networks-based Model for Sales Prediction", In Proceedings of the, International Conference on Artificial Intelligence, Robotics and Control, 2019, pp. 61-67.

[18] A. Sbrana, A.L.D. Rossi, and M. C. Naldi, "N-BEATS-RNN: Deep learning for time series forecasting", In 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 765-768.

[19] Global Gaming Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026)

[20] IAB internet advertising revenue report. Available online: https://www.iab.com/wp-content/uploads/2019/05/Full-Year-2018-IAB-Internet-Advertising-Revenue-Report.pdf (accessed on 10 September 2021)

[21] Y. Ju, X. Wang, and X. Chen, "Research on OMR Recognition Based on Convolutional Neural Network Tensorflow Platform", 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2019, pp. 688-691.

[22] A. Samraj, D. Sovmiya, K. A. Deepthisri and R. Oviya, "Food Genre Classification from Food Images by Deep Neural Network with Tensorflow and Keras", Seventh International Conference on Information Technology Trends (ITT), 2020, pp. 228-231.

[23] A. Kolte, B. Mahitha and N.V.G. Raju, "Stratification of Parkinson Disease using python scikit-learn ML library," International Conference on Emerging Trends in Science and Engineering (ICESE), 2019, pp. 1-4.

[24] S. Van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," in Computing in Science Engineering, vol. 13, no. 2, March-April 2011, pp. 22-30.

[25] N. Ari and M. Ustazhanov, Matplotlib in python," 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1-6.

[26] S. G. Iyer and A. D. Pawar, "Machine Learning Model for Predicting Price of Processors using Multivariate Linear Regression," International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019, pp. 52-56.

[27] S. Shariar and K. M. Azharul Hasan, "GPU Accelerated Indexing for High Order Tensors in Google Colab," IEEE Region 10 Symposium (TENSYMP), 2020, pp. 686-689.

[28] L. Wu and Y. Wang, "Modelling DGM(1,1) under the Criterion of the Minimization of Mean Absolute Percentage Error," Second International Symposium on Knowledge Acquisition and Modeling, 2009, pp. 123-126.

[29] C. Reyes, T. Hilaire, S. Paul and C. F. Mecklenbräuker, "Evaluation of the root mean square error performance of the PAST-Consensus algorithm," 2010 International ITG Workshop on Smart Antennas (WSA), 2010, pp. 156-160.

[30] F. H. Fard and B. H. Far, "Detecting a certain kind of emergent behavior in multi agent systems applied on mase methodology," 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2013, pp. 1-4.