Detection of Balance Anomalies with Quantile Regression: the Power of Non-symmetry

David Muelas* †
david.muelas.recuenco
BBVA Data & Analytics

Luis Peinado*
luis.peinado
BBVA Data & Analytics

Axel Brando*
axel.brando
BBVA Data & Analytics
Universitat de Barcelona

Jose A. Rodríguez-Serrano*
joseantonio.rodriguez.serrano
BBVA Data & Analytics

ABSTRACT

The analysis of end-of-day account balance offers key indicators of customers' situation and can help to define advisory and proactive actions towards their financial well-being. The detection of unexpected variations in its evolution arise as a key matter, given that they may expose events that require immediate attention or substantial changes in customers' context. We present a system that puts together (i) time series forecasting with uncertainty; (ii) statistical detection of extreme values; and (iii) analysis of financial transactions to notify and improve the visibility of this type of events for customers. We discuss the main details of the models and data engines that encompass this system, and present empirical results to illustrate their outcomes in real use cases.

CCS CONCEPTS

• **Information systems** → **Data analytics**; • **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

balance forecasting, quantile regression, anomaly detection, node embedding

ACM Reference Format:
David Muelas, Luis Peinado, Axel Brando, and Jose A. Rodríguez-Serrano. 2020. Detection of Balance Anomalies with Quantile Regression: the Power of Non-symmetry. In *KDD Workshop on Machine Learning in Finance, August 24, 2020, San Diego, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/-

*Authors contributed equally to this research. Email domain: @bbvadata.com .
†Corresponding author.

KDD-MLF-2020, August 24, 2020, San Diego, CA, USA

ACM ISBN -…$15.00
https://doi.org/-

1 INTRODUCTION

The end-of-day account balance offers several key indicators of customers' financial situation. Moreover, the dynamic evolution of this parameter can help to characterize customers' behavioral patterns and depict advisory and proactive actions towards their financial well-being.

In this regard, forecasting the future values of end-of-day account balance can help to extract insights to reduce the impact of undesirable situations—i.e., negative balance or lacks of liquidity—and determine whether an unexpected event may have happened—i.e., excursions in balance amounts related to values lying out of likely thresholds. Needless to say, these tasks can have a direct impact on customers' quality of experience as they can reduce their exposure to disappointing situations—such as payment of fees or late detection of undesired abrupt changes in personal accounts' balance.

On its basic foundations, solving this problem is intrinsically tied to time series forecasting. This is a broadly studied topic given its prominent importance for many problems in diverse scopes [9], which explains the literature devoted to techniques and applications. Early attempts to model and extract patterns from time-varying data relied on statistical models such as ARMA and ARIMA. The adjustment of this type of models was integrated into classical analytical frameworks—such as Box-Jenkins methodology or other data-driven approaches based on careful transformations of input data [20]—to improve the process of fitting and quality of results.

However, two key matters arise when considering the particularities of customers' balance series. First, these series can present complex patterns, with seasonal components and periodic events. Second, these patterns are related to the financial context of customers—that is, their usage of accounts and diverse expenses and income sources—which generates a broad heterogeneity in the observable trends. Therefore, the forecasting techniques applied to this type of series should be able to capture this variability.

Neural Networks (NN) were soon identified as a promising model family given their flexibility and adaptability to complex patterns [23]. Since the early explorations of NN in the field of time series forecasting, there have appeared several architectures and learning approaches to capture the peculiarities of these data—for instance, the Long Short-Term Memory (LSTM) architecture [13]. These models have proven to better suit application contexts that involve humongous data amounts and heterogeneous series.

However, and generally speaking, time series forecasting solutions have focused on the modeling and prediction of point-wise estimates, while the stochastic nature of processes calls for the introduction of *probable* future values. That is, as there are several sources of uncertainty [10], forecasting models should include mechanisms to account for the confidence and dispersion of the regression results. Moreover, this can also simplify the application of forecasting results to anomaly detection tasks, as it enables the comparison of the observed values with the predicted ones at different confidence levels.

With these principles in mind, we have implemented a complete system that supports the detection and analysis of anomalous events in customers' account balance. This system encompasses three stages to determine whether a variation in the end-of-day balance amount is out of a confidence band and select a set of financial transactions that relate to those events. The main contributions of this work are threefold:

- First, we present a deep learning regression model that has proven its capabilities to foresee customers' end-of-day balance time series. This model is able to estimate the distribution of future values by conducting a quantile regression on time series using a recurrent neural network.
- Second, we describe a method based on the predicted quantiles to detect abrupt excursions in such series. This method is based on probability thresholds to detect significant deviations in series.
- Third, we show a method to link the extreme events in balance time series to the underlying financial transactions, in order to gain insights into the type of situations that generate such extreme events.

To present these contributions, the rest of the paper is organized as follows. Section 2 reviews several previous works that frame our solution and provided the theoretical foundation of our analysis. Then, Section 3 details the main components of our solution and the relation between them. On its part, Section 4 presents the use case in which we have applied this solution, and details how we have conducted the performance evaluation of each component. With this, Section 5 focus on the empirical results of the system and discuss their implications. Finally, Section 6 summarizes the key remarks and concludes this work.

2 RELATED WORK

As stated above, there are three main topics that intersect in this work: time series forecasting, anomaly detection, and analysis of financial transactions. In the following, we review some previous results in those fields that relate to our solution.

Time series forecasting is an extensively explored field of study [4]. The proposals and techniques have spanned very diverse perspectives [9], from the use of classic statistical tools—like exponential smoothing [12] or auto-regressive methods [20]—to more complex models such as NN [7, 13, 23] or Support Vector Machines (SVM) [6, 17]. Particularly, NN models have proven to be quite successful for the modeling of complex trends in time series when there are diverse series typologies, and offer a complete framework to obtain probabilistic estimates of future values.

In this regard, several previous works made use of different approaches to forecast the future distribution of values in time series. On the one hand, one of the multiple trends to cope with this matter focuses on learning the parameters of the distribution, using either a specific family such as Gaussian distributions [22, 24] or more flexible representations with finite [3] or continuous mixture models —such as the case of UMAL [5]. On the other hand, other solutions have applied quantile regression [18, 19] to determine probabilistic levels for future values of series, sometimes in combination with deep methods [8, 27]. Our solution follows this latter approach, offering a grid of quantiles at the output.

Remarkably, common probabilistic outlier detection techniques [1] can be used after forecasting future distributions for time series. This is the basis of the second stage in our solution, in line with previous works that defined algorithms for the detection of extreme events using confidence intervals [28]. In our case, we operate with order statistics to define bands that relate to values with low probability w.r.t. the predicted distribution, using the grid of quantiles produced by the forecasting stage.

Finally, as stated above the detection of anomalous events may render useless if customers do not receive a set of explanations of the causes behind the events. To better understand these causes, we have applied representation learning to the financial transactions that occur during the anomalous events.

In line with the state of the art, we have accomplished this process using graph analysis and node embedding techniques [16]. We have focused on techniques such as DeepWalk [21] and node2vec [14], as they suit the detection of *communities* or clusters among nodes in the graph. The analysis of these clusters have followed some of the principles that are usually applied during the evaluation of embedding methods' results [2, 25], as we have analyzed the relation of the original movements' features with the nodes' representations.

3 ANALYTIC SOLUTION

Hereinafter, we present the main details of our analytic solution. Specifically, we discuss the main stages along the pipeline depicted in Fig. 1. Namely, we first describe the solution applied to the forecasting with uncertainty of time series (**S1** in the diagram); then we discuss the strategy to determine atypical observations for the punctual value of such series (**S2**); and finally we introduce the key ideas of the analysis of the causes of atypical observations (**S3**).

3.1 Forecasting of financial time series

The first stage in our system relies on a deep learning model that is trained to forecast the quantiles of net balance daily time series. This approach is able to capture complex patterns, determining likely future trajectories from the past values of series. Specifically, we used an iterated recurrent neural network—see the schematic representation in Fig. 2. Here, x_c denotes the input (i.e. the balance datum) to the network at a given time step, $\phi(\cdot)$ denotes the deterministic function implemented by the trained recurrent cell, and o_{c+1} is the output of the recurrent cell. Next, we describe the proposed training process, which is a variation of RNN aimed at achieving long-term forecasting robustness.

Our modification upgrades a RNN to achieve long-term forecasting robustness by using two stages in the unfolding of the RNNs. We

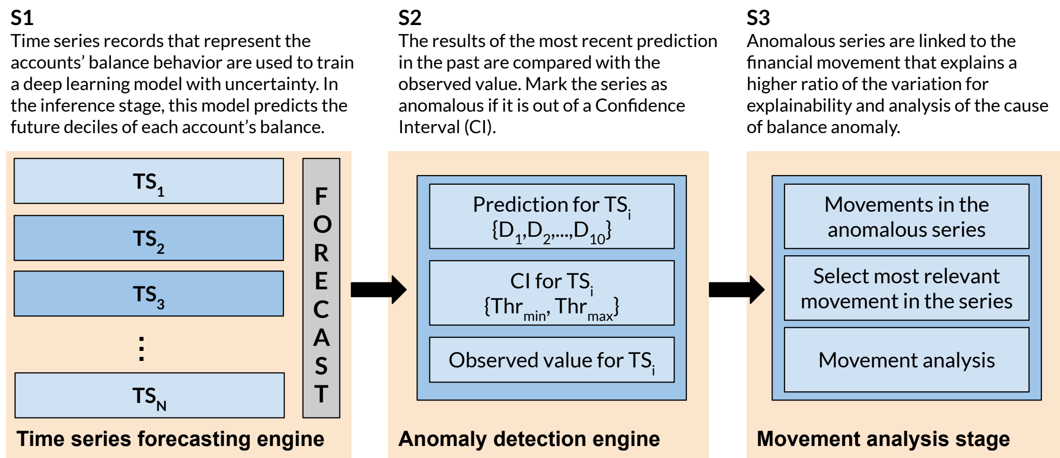


Figure 1: Architectural diagram of the solution. We distinguish three main stages: (S1) time series forecasting with uncertainty; (S2) detection of anomalous series; (S3) financial movement analysis.

consider working with series of length $N + K$. The first stage uses the first N input data. Each time, the input to the series is the newly observed input datum x_c at a given time. The series outputs o_{c+1}, which should approximate the next input datum x_{c+1}. This stage is common in RNN training, and we denote it as *Confidence*. The next phase considers K more inputs. The difference in this phase is that each predicted output o_{c+t} is fed directly to the next input, replacing x_{c+t}, which is not used as input. However, the output is still compared with x_{c+t} to compute the loss. In other words, we are forcing the recurrent cell to learn to forecast correctly K time steps in advance. This phase is denoted as *Trust*. Once the model is trained, given an input series, any arbitrary number of future time steps can be generated. This part is denoted as *Hope*.

However, in order to do long-horizon forecasting and ensure that the forecast of future steps is meaningful, the RNN predictions alone are not enough. Next, we describe how we introduce management of uncertainty in our system. We seek for methods to model forecasting uncertainty with three properties: (1) each prediction should carry some sort of input-dependent 'prediction interval' (2) the expected behavior for uncertainty is to increase with longer forecasting horizons, and (3) the uncertainty quantities should be calibrated with data.

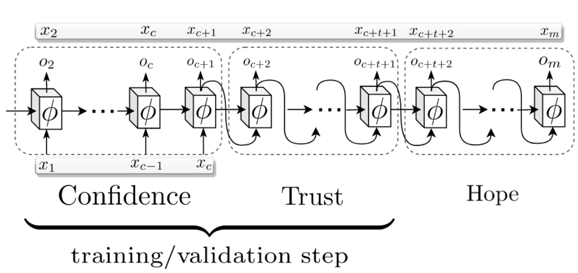


Figure 2: Schematic representation of the iterated regression model applied to time series forecasting.

In order to model the uncertainty of the time series, we selected an approach based on quantile regression [19] after testing different methods—such as Uncertainty Generalized Normal Distribution [5] or Montecarlo Drop Out [11]. Differently from most regression tasks, which model the *expected value* of the target variable given the input [26], quantile regression estimates *a desired quantile* of the target variable given the input. This is achieved using the following loss function, sometimes known as pinball loss or quantile loss:

$$\mathcal{L}_\tau(y, x) = (y - f_z(x)) \cdot (\tau - 1[y < f_z(x))]) \quad (1)$$

Here, (x, y) is a training sample and $f_\tau(x)$ is the function implemented by the forecaster. One can show that, after training, $f_\tau(x)$ approximates the τ^{th} quantile of $p(y|x)$. For illustration, Fig. 3 displays the result of quantile regression on a synthetic dataset. Here, $f_\tau(x)$ has been implemented with a 3-layer dense neural network, and designed to output the quantiles $\tau = 0.1$, $\tau = 0.5$ and $\tau = 0.9$. We see how, for each x, the forecast of $f_{0.5}$ approximates the median of y, and $f_{0.1}$ and $f_{0.9}$ adjust to the scale of the data; as expected, effectively informing about prediction intervals.

Coming back to our recurrent network, if we consider the loss of predicting the value of a certain quantile $q_{\tau_{i-1}}$ and evaluate the loss with respect to the next observation x_i, the loss function becomes:

$$\mathcal{L}_\tau(x_i, q_{\tau_{i-1}}) = (x_i - \phi(q_{\tau_{i-1}})) \cdot (\tau - 1[x_i < \phi(q_{\tau_{i-1}})]). \quad (2)$$

The reason of the subscript τ is that we might be interested in predicting the values of several quantiles simultaneously, for instance for $\tau = 0.1$, $\tau = 0.5$ and $\tau = 0.9$. This can be achieved by combining all the τ-dependent losses into a single loss function, e.g through averaging:

$$\mathcal{L}(x_i, \mathbf{Q_{i-1}}) = \sum_\tau \mathcal{L}_\tau(x_i, q_{\tau_{i-1}}), \quad (3)$$

as done e.g. in [8], where $\mathbf{Q_{i-1}}$ is the vector of dimension $|\tau|$ holding all the quantile estimates, and typically the networks $\phi(q_{\tau_{i-1}})$ would share some parameters between the different τs. Indeed, since each recurrent cell takes as input (and outputs) $|\tau|$ quantile estimates,

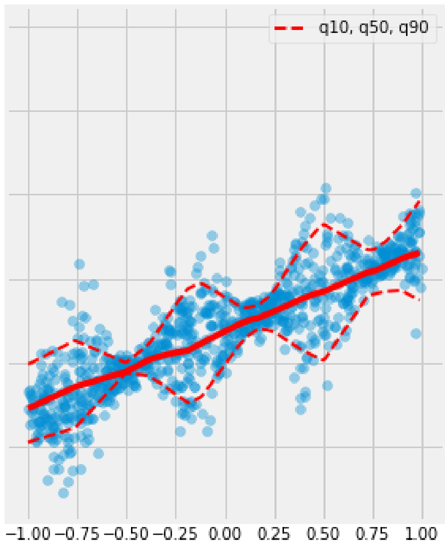


Figure 3: Example of quantile regression on a syntehtic dataset, showing the forecast of the quantile $\tau = 0.5$ (solid line) and $\tau = 0.1$ and $\tau = 0.9$ (dashed lines), for each x.

and holds (and updates) a hidden state $\mathbf{h_i} \in \mathbb{R}^s$, the function to learn for each recurrent cell has the form

$$\begin{aligned} \phi:\ \mathbb{R}^{|\tau|} \times \mathbb{R}^s &\to \mathbb{R}^{|\tau|} \times \mathbb{R}^s \\ (q0.05_i, q0.5_i, q0.95_i, h_i) &\mapsto (q0.05_{i+1}, q0.50_{i+1}, q0.95_{i+1}, h_{i+1}) \end{aligned} \quad (4)$$

Remarkably, one of the key aspects is that this network architecture with estimations of quantiles allows us to model the balance time series in a very flexible way. Specifically, the quantile regression is able to capture the non-symmetric nature of these series.

3.2 Detection of anomalies

We use a threshold-based anomaly detection approach that relies on the quantile regression results. Specifically, our engine raises an alarm if

$$\begin{aligned} (\mathbb{P}\{Balance(t) < X_t\} < Thr_{min}) \vee \\ (\mathbb{P}\{Balance(t) > X_t\} > Thr_{max}) \end{aligned} \quad (5)$$

where $X_t, Balance(t)$ are the predicted quantiles and observed values for time t respectively, and $(Thr_{min} + Thr_{max})$ is the anomalous range's probability. To offer noticeable results without overwhelming customers, we heuristically control $(Thr_{min} + Thr_{max})$ by considering as anomalies those observations outside an interval of the form:

$$\begin{aligned} [q_{0.05} - k_1 \frac{(q_{0.50} - q_{0.05})}{(q_{0.95} - q_{0.05})}(q_{0.95} - q_{0.05}), \\ q_{0.95} + k_2 (1 - \frac{(q_{0.50} - q_{0.05})}{(q_{0.95} - q_{0.05})})(q_{0.95} - q_{0.05})] \end{aligned} \quad (6)$$

which is inspired in the proposal by Hubert and Vandervieren [15] to adapt the well-known Tukey criterion to asymmetric distributions.

Here, both k_1 and k_2 enable the introduction of differential weights for abnormal values above / below the non-atypical values' range. Note that, as Hinkley's coefficient:

$$0 = \frac{((q_{0.95} - q_{0.50}) - (q_{0.50} - q_{0.05}))}{(q_{0.95} - q_{0.05})} \quad (7)$$

is 0 for symmetric distributions, if we scale $[q_{0.05}, q_{0.95}]$ to $[0, 1]$, then

$$|q_{0.50} - q_{0.05}| + |q_{0.95} - q_{0.50}| = 1 \quad (8)$$

and

$$\begin{aligned} |q_{0.50} - q_{0.05}| \to 0.5 \\ |q_{0.95} - q_{0.50}| \to 0.5 \end{aligned} \quad (9)$$

when the distribution tends to be symmetric.

Therefore, with $k_1 = k_2 = 4$, these intervals converge to the following expression for fairly symmetric distributions:

$$[q_{0.05} - 2(q_{0.95} - q_{0.05}), q_{0.95} + 2(q_{0.95} - q_{0.05})] \quad (10)$$

which empirically showed good results in terms of detected events—although it may be somehow restrictive in case of Gaussian distributions, it suits the heavier-tailed nature of balance series and controls the number of alerts when asymmetry increases.

Anyhow, these parameters can be tuned to reflect a less restrictive behavior or to introduce bias towards the detection of negative or positive divergences.

3.3 Analysis of transactions behind events

The stages described so far enable the system to pinpoint extreme events, but their outcomes may be too narrow to offer meaningful and self-explanatory insights about them. In this light, we have complemented the detection of abrupt changes in the balance series with the analysis of transactions related to such events.

To do so, the last stage of the pipeline determines whether there is a small number of movements that can explain the change of balance. In other words, once an anomalous event is detected in the end-of-day balance for account k at day t, this module searches in the set $\mathcal{M}_t^k$ of movements for that account and day, and assesses if there exists a subset $\mathcal{S}_t^k$ such that:

$$\begin{cases} \mathcal{S}_t^k \subseteq \mathcal{M}_t^k \\ |\mathcal{S}_t^k| \leq N \\ |\sum_{m \in \mathcal{S}_t^k} m| \geq \text{Thr} \cdot |\Delta\text{Balance}^k(t)| \end{cases} \quad (11)$$

where N, Thr are parameters to control the maximum number of movements to be considered in the explanation of the anomalous event and the minimum ratio of the variation that must be covered, respectively.

However, there may be different typologies of events behind the anomalous values in balance series that require differentiated treatments. We defined a graph-based exploration of the movements to assess the existence of these typologies via the detection of *communities*.

For the movements in the union of $\mathcal{S}_t^k$ for all the accounts, this engine selects a set of features—e.g., amount or category of expense / income—and transforms any continuous feature into categorical ones—e.g., amount to an absolute indicator of its magnitude. This representation of movements allows us to define a bipartite graph consisting of nodes that represent transactions and nodes that represent the levels of their features.

Once we have such a graph representation, we have applied node2vec [14] to obtain a node embedding and determine whether there are clusters of transactions that relate to specific sets of movements' attributes. To do so, we have (i) used clustering algorithms (K-means) on the vectors corresponding to transactions in the embedding, and (ii) analyzed how the transactions' features relate to the resulting clusters.

4 METHODOLOGY AND EXPERIMENTS

In this section we describe the context and testing of our system. To do so, on the one hand we briefly depict the use case in which we applied it; and on the other hand we define the data and methods to assess the system's performance and behavior.

4.1 Definition of the use case

As stated in the introduction, all the elements described above are part of an automatic system for the detection of unexpected changes in customers' balance time series. With this, customers could receive notifications that include details about such events, to increase their awareness about the situation. To fulfill this behavior, the solution was deployed as follows:

(1) The time series forecasting engine is in charge of the production of balance predictions with uncertainty. With this, it offers a basis to foresee future values together with a level of probability of occurrence.
(2) On its part, the anomaly detection engine takes as input the most recent forecasts of balance quantiles for the day of execution—i.e., *the most recent forecast for the present*. Therefore, the application of the statistical criterion detects extreme values w.r.t. the quantiles that were predicted for the present.
(3) The extreme events detected in the time series are linked to the financial transactions happening between the production of forecasts and the detection of the anomalies. These transactions are clustered in terms of some of their attributes, to gain insights into the type of situations that generate the abrupt and anomalous values in the series.

4.2 Description of methods

The results that we present in the following sections are mainly obtained from the application of the previous elements to real financial data in a pre-productive environment. We have also included some synthetic modifications of the input data to illustrate and validate relevant characteristics of our system.

Therefore, our methodological approach was somehow similar to a *natural experiment*: that is, after the validation of the engines and models in the system, we focused on the analysis of its output after processing data from productive scenarios. We believe that, although this approach may make harder the extrapolation of results, it offers a more realistic and interesting evaluation in an industrial environment.

Our data sources cover financial transactions and end-of-day balance time series from anonymized customers. Specifically, the input for the forecasting engine are series lasting several months [1], and its output consists of the values of quantiles for the next days—the specific number of days is a settable parameter, and aleatoric uncertainty increases when forecasts are made for several weeks in the future.

[1]Details regarding this matter cannot be disclosed.

We have included results for a random day of late 2019 for the behavioral characterization of the anomaly detection engine. The time lapse between quantile forecasts and the application of the anomaly detection engine was one day.

Finally, the analysis of movements related to the most relevant financial transactions within anomalous periods was restricted to non-sensitive features of transactions—namely, the category of expense / income; amount of the movement; and movement typology.

Regarding the tests with synthetic variations of the input data, we added some common transformations to balance time series to illustrate and assess their effect on the forecasting engine output. Specifically, we focused on the effects of (i) linear trends, (ii) adding constant values and (iii) increasing series' variance, to determine how forecasts can adapt to typical situations in our scope.

5 RESULTS

5.1 Forecasting

We first focus on the performance of the time series forecaster. To do so, we consider its results in a task to characterize its precision and calibration; and then illustrate its qualitative response during inference to typical variations of the input time series.

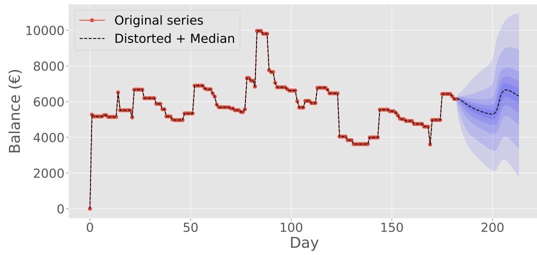
Table 1 summarizes the performance of the time series forecasting model in a supervised task. Specifically, the prediction of overdraft in account balance in next 14 days by using different quantiles. We consider overdraft prediction when a determinate quantile—[0.6,0.7,0.8,0.9,0.95]—is less than zero. These results show the good calibration of the algorithm, given that the accuracy is near of the quantile value—i.e., the relative frequency of overdraft is near the probability of the balance being less or equal than zero.

Table 1: Overdraft prediction considering 6 different data points as samples to evaluate the performance of each quantile during the next 14 days.

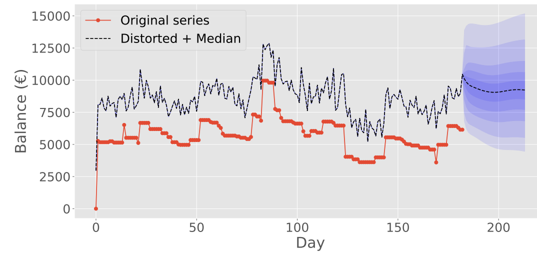
Precision Table	0.6	0.7	0.8	0.9	0.95
Minimum sample	6487	2857	766	148	49
Maximum sample	44,109	30,847	20,393	10,294	3649
Accuracy	72.01±6.38	75.52±7.01	82.07±6.286	88.59±6.18	89.19±5.25

Finally, Fig. 4 presents examples of the output of this model—observed values are presented in red with dots, predicted quantiles are shadowed, and median value is displayed with a black dashed line. It also includes the results when the original data is transformed adding noise and trends—namely, Gaussian noise with $\mu = 0.5 \cdot avg(X)$, $\sigma = 0.5 \cdot std(X)$ being X the original time series; a constant value (5000) plus a linear trend $50 \cdot t$; and a constant value (5000) plus a linear trend $100 \cdot t$.

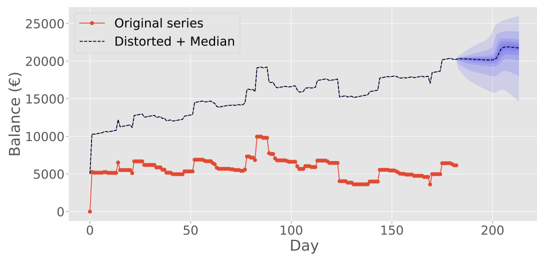
These synthetic experiments are intended to show (i) the adaptability of the forecast to different trends; and (ii) the increasing uncertainty as a result of higher input variability and longer prediction intervals. These effects clearly appear in the inference results, and illustrate the flexibility of this approach for sets of time series



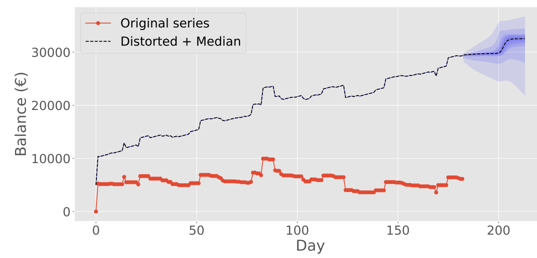
(a) Original time series.



(b) Original time series plus Gaussian noise.



(c) Original time series plus linear trend, $50 \cdot t$.



(d) Original time series plus linear trend, $100 \cdot t$.

Figure 4: Example of pattern learning in the output of the time series forecasting engine when several transformations are applied to a real balance time series.

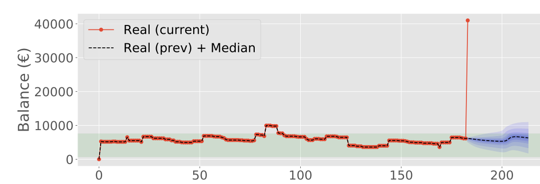
that include heterogeneous patterns and typologies in terms of trends and variability.

5.2 Spotting anomalies in financial time series

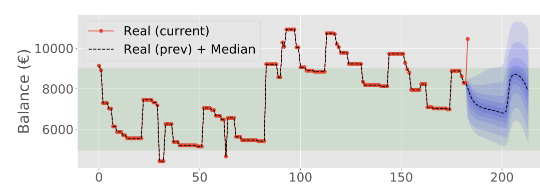
The next step in our system is the anomaly detection engine. As stated above, it relies in the most recent forecast of time series distribution to evaluate how likely is their observation. Some illustrative examples are included in Fig. 5, which presents observed values, forecasts, and confidence bands for a couple of series. Note that, although the bands are considered only for the detection of anomalous events that happened in the last observed day, we have included them along the complete series to assess how they relate to past values.

The anomaly detection engine detects the observable spikes in the series—last observation of real values, which is displayed as a red line with dots—relying on the forecasts of the previous day—dashed black line and shadowed regions. Here, we also included the confidence bands—regions in green along each plot—showing how these points correspond to values out of them. These illustrative examples show how the bands (and the predicted quantiles) reflect the asymmetry of the series and, therefore, allow the engine to detect values which are unlikely following the expected distribution for future values of the series.

Regarding the evaluation of the outcomes of the anomaly detection engine, we have measured if the detected events relate to abrupt changes in the balance series—hence, following an unsupervised approach. Fig. 6 illustrates the relation between the absolute change in the series and the minimum distance to the confidence band and their respective marginal densities.



(a) Low relative variance, broad confidence interval.



(b) High relative variance, narrow confidence interval.

Figure 5: Example of time series with anomalous event. The last observation is outside the projected confidence interval (in green).

This analysis shows that there is a noticeable relation between these two indicators, which is necessary to guarantee that detected anomalous events are significant. However, low changes can be also detected if they appear in very stable series or if the predicted trend is far from the observed value. Note that this behavior can be qualitatively assessed in the second example in Fig. 5, where the forecasts captured the decreasing trend in the series while the actual value exhibited a fairly atypical increment in balance.

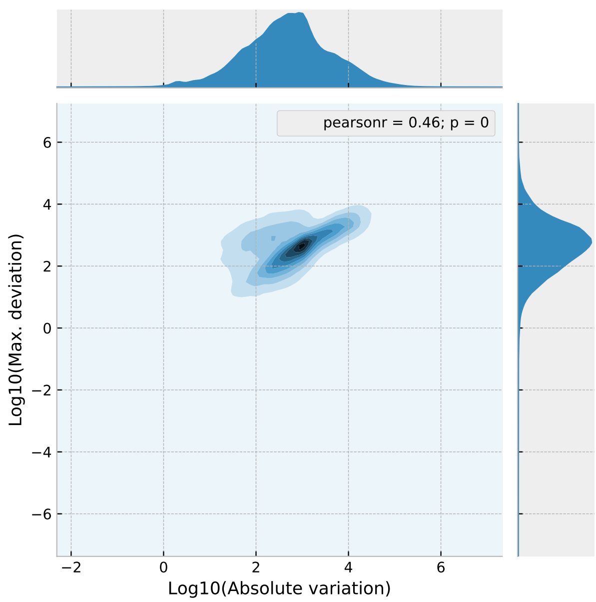


Figure 6: Analysis of the relation between the deviation (minimum distance from the confidence threshold) and the absolute change in the time series.

5.3 Analysis of the movements

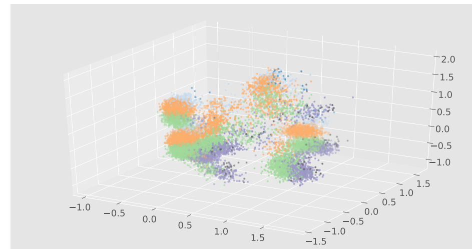
Finally, we now focus on the outcomes of the analysis of movements behind the anomalous events in the series. We recall that we have considered the possible clusters or communities among the movements by analyzing the relations in a bipartite graph using node2vec.

Fig. 7 shows the results of this process. There, we present the representation of movements after the application of PCA to the vectors of the node embedding. Colors relate to the values of some of the financial transactions' features—specifically, (a) the rounded decimal logarithm of the absolute movement amount; (b) the expense / income category; and (c) the type of movement.

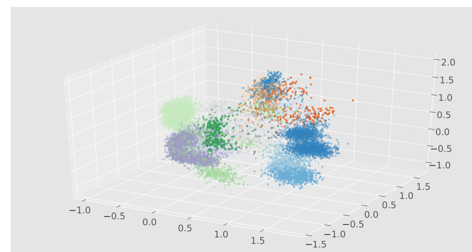
This analysis shows that this process is able to capture and differentiate groups of movements that share part of their features. Hence, this approach can offer a basis to define custom policies for anomalous events related to different movement typologies.

5.4 Discussion

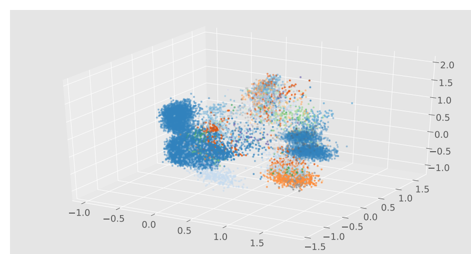
We have presented several results regarding the application of multiple state-of-the-art machine learning techniques to a complete use case for customer advising in banking services. We would like to remark three lessons learned from this work, which we believe can help to pave the way for future solutions with further added value for customers:



(a) Amount magnitude.



(b) Expense / income category.



(c) Type of movement.

Figure 7: Clustering of anomalous events w.r.t. (a) amount; (b) expense / income category; (c) type of movement.

(1) We have seen how the modeling of uncertainty in real applications of machine learning offers outstanding qualitative improvements for the analysis of forecasts. In this regard, our solution provides means to (i) capture complex time series behaviors and (ii) observe and quantify the possible variability of future values w.r.t. the model predictions.
(2) We have illustrated how this approach provides a natural basis to build confidence intervals to both evaluate the proper model predictions and detect excursions of future observations—i.e., those with a very low probability of occurrence.

(3) This system is intended to offer automatic notifications to end users. Therefore, we have included a final step which is able to pinpoint causes of the anomalous event in the balance series. Furthermore, this can offer further information to enrich the context via the analysis of anomaly groups.

We believe that these lessons can help researchers and practitioners in the machine learning and banking communities during the definition of future services and use cases.

Anyhow, we envision some aspects that may be tackled to improve this type of systems. Particularly, the spotting and information provided by this type of solutions may be also related to customers' preferences and overall interests. Therefore, we foresee the introduction of feedback and user-centered information to not only notify the apparent occurrence of unexpected events but also to include an explicit optimization of utility for customers.

6 CONCLUSIONS

Along this work, we have presented the results of the application of a complete analytical pipeline that offers comprehensive insights for customers regarding their end-of-day account balance. To accomplish this task, our solution puts together time series forecasting (S1); anomaly detection techniques (S2); and analysis of causes behind the detected anomalous events (S3).

Our results illustrate the potential of advanced analytic to help customers with the control of their financial situation. Future work lines include the refinement of the analysis by means of customization of results by introducing reinforcement w.r.t. interest metrics.

ACKNOWLEDGMENTS

We would like to especially thank to the people in both the Advice and Cross Analytics Programs at BBVA Data & Analytics for their support and help. This is extensive to the rest of colleagues at the BBVA AI Factory, as these results are part of the outstanding team work.

REFERENCES

[1] Charu C Aggarwal. 2017. Probabilistic and statistical models for outlier detection. In *Outlier Analysis*. Springer, 35–64.

[2] Amir Bakarov. 2018. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536* (2018).

[3] Christopher M Bishop. 1994. Mixture density networks. (1994).

[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[5] Axel Brando, Jose A Rodriguez, Jordi Vitria, and Alberto Rubio Muñoz. 2019. Modelling heterogeneous distributions with an Uncountable Mixture of Asymmetric Laplacians. In *Advances in Neural Information Processing Systems*. 8836–8846.

[6] Li-Juan Cao and Francis Eng Hock Tay. 2003. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks* 14, 6 (2003), 1506–1518.

[7] Yagmur Gizem Cinar, Hamid Mirisaee, Parantapa Goswami, Eric Gaussier, and Ali Aït-Bachir. 2018. Period-aware content attention RNNs for time series forecasting with missing values. *Neurocomputing* 312 (2018), 177–186.

[8] Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. 2018. Implicit Quantile Networks for Distributional Reinforcement Learning. In *International Conference on Machine Learning*.

[9] Jan G De Gooijer and Rob J Hyndman. 2006. 25 years of time series forecasting. *International journal of forecasting* 22, 3 (2006), 443–473.

[10] Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? Does it matter? *Structural safety* 31, 2 (2009), 105–112.

[11] Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. arXiv:stat.ML/1506.02142

[12] Sarah Gelper, Roland Fried, and Christophe Croux. 2010. Robust forecasting with exponential and Holt–Winters smoothing. *Journal of forecasting* 29, 3 (2010), 285–300.

[13] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 28, 10 (2016), 2222–2232.

[14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[15] Mia Hubert and Ellen Vandervieren. 2008. An adjusted boxplot for skewed distributions. *Computational statistics & data analysis* 52, 12 (2008), 5186–5201.

[16] Anish Khazane, Jonathan Rider, Max Serpe, Antonia Gogoglou, Keegan Hines, C Bayan Bruss, and Richard Serpe. 2019. DeepTrax: Embedding Graphs of Financial Transactions. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 126–133.

[17] Kyoung-jae Kim. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55, 1-2 (2003), 307–319.

[18] Roger Koenker and Kevin F Hallock. 2001. Quantile regression. *Journal of economic perspectives* 15, 4 (2001), 143–156.

[19] Roger W Koenker and Gilbert Bassett. 1978. Regression Quantiles. *Econometrica* 46, 1 (1978), 33–50. https://EconPapers.repec.org/RePEc:ecm:emetrp:v:46:y:1978:i:1:p:33-50

[20] Spyros Makridakis and Michele Hibon. 1997. ARMA models and the Box–Jenkins methodology. *Journal of Forecasting* 16, 3 (1997), 147–163.

[21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[22] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* (2019). https://doi.org/10.1016/j.ijforecast.2019.07.001

[23] Zaiyong Tang, Chrys De Almeida, and Paul A Fishwick. 1991. Time series forecasting using neural networks vs. Box-Jenkins methodology. *Simulation* 57, 5 (1991), 303–310.

[24] Bin Wang, Jie Lu, Zheng Yan, Huaishao Luo, Tianrui Li, Yu Zheng, and Guangquan Zhang. 2019. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[25] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C Jay Kuo. 2019. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing* 8 (2019).

[26] Larry Wasserman. 2010. *All of Statistics: A Concise Course in Statistical Inference.*

[27] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2017. A multi-horizon quantile recurrent forecaster. In *NIPS Workshop on Time Series*.

[28] Lingxue Zhu and Nikolay Laptev. 2017. Deep and confident prediction for time series at Uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 103–110.