TimeFork: Interactive Prediction of Time Series

Sriram Karthik Badam,[1] Jieqiong Zhao,[2] Shivalik Sen,[3] Niklas Elmqvist,[1] and David Ebert[2]

[1]University of Maryland
College Park, MD, USA
{sbadam|elm}@umd.edu

[2]Purdue University
West Lafayette, IN, USA
{zhao413|ebertd}@purdue.edu

[3]Birla Institute of Technology and Science
Goa, India
shvlksen@gmail.com

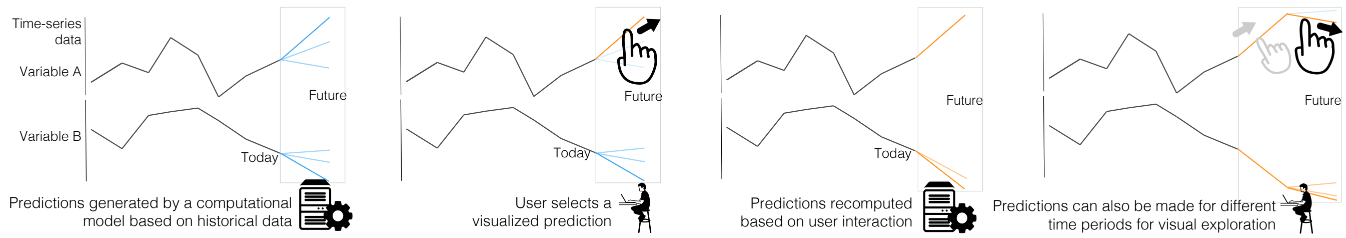

Figure 1. TimeFork is a technique for interactive prediction of time-series data. It uses computational models to create and show predictions on time-series visual representations. They are explored through a dialogue, driven by interaction, to see how predictions for time-series variables change based on others. This approach harnesses user knowledge of external factors and the ability of computational models to predict based on past trends.

ABSTRACT

We present TimeFork, an interactive prediction technique to support users predicting the future of time-series data, such as in financial, scientific, or medical domains. TimeFork combines visual representations of multiple time series with prediction information generated by computational models. Using this method, analysts engage in a back-and-forth dialogue with the computational model by alternating between manually predicting future changes through interaction and letting the model automatically determine the most likely outcomes, to eventually come to a common prediction using the model. This computer-supported prediction approach allows for harnessing the user's knowledge of factors influencing future behavior, as well as sophisticated computational models drawing on past performance. To validate the TimeFork technique, we conducted a user study in a stock market prediction game. We present evidence of improved performance for participants using TimeFork compared to fully manual or fully automatic predictions, and characterize qualitative usage patterns observed during the user study.

Author Keywords

Visual Analytics; Human-in-the-loop; Time Series; Visual Prediction; User Study

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Interaction styles*; I.3.6 Computer Graphics: Methodology and Techniques—*Interaction techniques*

CHI'16, May 07-12, 2016, San Jose, CA, USA

DOI: http://dx.doi.org/10.1145/2858036.2858150

INTRODUCTION

Time-series data is prevalent across many domains, including financial markets, healthcare, meteorology, seismology, and astronomy. Information visualization (InfoVis) and visual analytics (VA) techniques are commonly used to analyze such time-series datasets. Recent techniques [52, 53] in these fields visualize time series and related statistical metrics to enable analysts to interactively reason about the temporal data. Their goal is to better leverage the human ability to gain understanding from visualizations, including identifying the underlying trends, anomalies, and correlations, through an interactive and undirected search [28]. However, going beyond understanding time-series data to predicting its future behavior remains a very challenging task, and few interactive exploration models exist that support this activity.

Prediction information has been integrated into time-series representations in the past [18]. Hao et al. [16] called this approach *visual prediction*: the act of visually predicting a time-series variable by observing the predictions from a computational model, shown alongside the time-series representations. However, traditional approaches cannot fully support visual exploration of future trends in complex multivariate datasets such as stock markets, weather, and healthcare data, mainly due to their lack of consideration of inter-variable relationships (e.g., if A increases, B decreases). Exploring these relationships through "what-if" questions (e.g., what if A increases?) can help us better judge the future than blindly trusting computational models that lack contextual information (e.g., legislation on fracking affecting oil stock prices).

In this paper, we present **TimeFork**, an interactive prediction technique for visual prediction of multivariate time series through visual exploration of predictions from computational models and the time-series data itself. To understand the motivation behind this technique, let us consider a potential use-case scenario involving prediction of the stock market. Stock traders have access to stock price and trade volume information, and they might also be following news, Twitter, and the earning reports of the companies they are interested in. Visualizations can aid them in understanding this data, e.g., how the stocks have changed over time, the current state of each stock, and how people are reacting on Twitter. However, knowing this information, they now have to make an estimate of the future for investment in each stock, even though they may not fully understand how to optimally account for their observations from the visualizations, the events happening during that time (such as product releases, mergers, and acquisitions), and the dynamic relationships among their stocks. By using a computer-supported technique such as TimeFork, traders can engage in an interactive dialogue with the computer to show them multiple predictions (with different confidences based on the historical data), account for different "what-if" scenarios (e.g., what if stock A increases due to their increased earnings), and come to a shared set of predictions with the computer through visual exploration.

The predictions generated by the computational models in TimeFork reflect the temporal (based on the past data of a variable) and conditional trends (based on the past relationships between variables), and are explored through an iterative process of judging what will happen to each variable based on specific future trends selected by the user for others (Figure 1). This interactive dialogue between the computer and the data analyst continues until the analyst finds sufficient information to make a decision. To showcase TimeFork, we developed STOCKFORK, a stock market VA application that instantiates the TimeFork technique using machine learning models trained on stock market data. Furthermore, we evaluated the effectiveness of TimeFork for visual prediction through a simulated stock trading experiment. In this study, participants were given an opportunity to invest virtual money on three stocks and make decisions based on their predictions. We found that the presence of a dialogue in TimeFork led to higher monetary gains compared to traditional automatic and manual prediction approaches in specific scenarios. The results of these analyses and our observations along with their implications are explained in later sections after descriptions of the related work, the technique, and the StockFork tool.

BACKGROUND

The motivation for TimeFork comes from the philosophy of visual analytics [46] itself: using computational models (for prediction) closely coupled with visual exploration (of time series). Below we discuss the literature related to time-series visualizations, time-series analytics, and visual prediction.

Time-Series Visualization

Heer et al. [19] provide examples of simple charts used for visualizing time-series data including index charts, horizon graphs, stacked graphs, and small multiples. More contextual representations for time-series include cluster and calendar-based visualizations [47], spiral visualizations [49] that map periodic sections of time-series into ring layouts, trend and trajectory visualizations [40] in financial markets, and multi-resolution layouts for handling overplotting in large time-series by switching between aggregated and detailed representations [17, 31]. Beyond static and interactive representations, there have also been animated representations to show trajectories of time-series variables. Moere et al. [35] built a 3D visualization based on information flocking [39] to show static and dynamic patterns arising in stock markets.

For large datasets, techniques [14] for visualizing the essence of the time series through perceptually important points—points that are most informative to visual identification—have been proposed. From an interaction standpoint, Hochheiser and Shneiderman's TimeSearcher [22] enabled dynamic quering of time-series visualizations through timeboxes—rectangular regions drawn on the 2D representations—to show details-on-demand. This tool was further developed to search time-series trends through similarity measures [6]. More recently, Zhao et al. [52] supported visual exploration of derived values and correlations within time-series data using lenses to transform time-series representations.

Time-Series Analytics and Prediction

Fu et al. [13] reviewed data mining models for time-series data including methods for representation, classification, segmentation, and pattern discovery. They identify two important pattern matching tasks: discovering frequent vs. surprising patterns. Approaches for motif discovery [9, 45], anomaly detection [8, 50], and novelty detection [33] answer these challenges. Of special interest to TimeFork are clustering and classification techniques that help to identify similar patterns across time series using distance measures [26] to quantify the relationships between time-series variables. Examples of such approaches [30] include using self-organizing maps [29], agglomerative clustering, and distance-based metrics. Rule mining [13, 21] to find rules defining the inter-variable relationships, is also of interest to computer-supported prediction techniques such as TimeFork.

Time-series prediction has often been attempted using statistical approaches such as regression analysis [18], as well as soft computing approaches such as neural networks, fuzzy logic, and evolutionary computation [37]. Among the latter, both supervised (e.g., multilayer perceptron) and unsupervised learning architectures (e.g., self-organizing maps [2]) have been applied. In financial markets, neural networks have been successful not only in predicting stocks based on their past price trends [48], but also based on the mood observed on Twitter [3]. However, stock prediction through automated techniques has mostly been restricted to individual stocks without generalizations. Beyond this, weather and healthcare data prediction has also relied on soft computing [32, 44].

Mixed-Initiative Techniques and Visual Prediction

Our TimeFork technique resembles mixed-initiative interaction [23], which targets a natural interleaving of contributions by people and computers to solve a task together [24]. These interaction techniques engage the user in an efficient dialogue with the system through interaction to resolve uncertainties. Mixed-initiative techniques have recently been used to analyze data on visual interfaces, where the user can interact with the parameter space of a computational model for classification to eventually come up with better results [27]. In visual analytics, semantic interaction [12] follows a similar ideology by coupling analyst's interaction with statistical models that incrementally learn and update visualizations (in their case, document layouts). Similar to these techniques, TimeFork aims at achieving an interactive dialogue between analysts and computers to make better predictions.

TimeSearcher3 [7] visualized predictions in a time-series visualization following a data-driven forecasting method. This system extrapolated time-series data by finding similar past sequences, while allowing the user to control the similarity metrics. Hao et al.'s [16] approach to supporting visual prediction went beyond TimeSearcher3 to support integration of multiple prediction models including autoregressive integrated moving average (ARIMA), Holt Winters (a seasonal method), and similarity-based models. They extended this approach to visualize peak-preserving predictions [18] for seasonal trends, interact with the model to adjust smoothing parameters, and connect predictions to similar past trends through brushing-and-linking. Malik et al. [34] presented a VA approach to explore correlations in multivariate spatiotemporal data. Still, visual prediction of multivariate time series—very common in financial, scientific, and medical domains—remains a challenge, due to the lack of support for visual exploration of possible futures across variables.

TIMEFORK: INTERACTIVE VISUAL PREDICTION

The TimeFork technique is designed to support interactive prediction of multivariate time-series data by integrating computational models with interactive visual analysis. Due to this inherent coupling with computational models, the design of TimeFork is based on the guidelines established by Horvitz [24] for mixed-initiative techniques including maintaining an efficient human-computer dialogue, transparently conveying the computer's capabilities [43], and preserving the working memory of the user during interaction. The basic TimeFork workflow is an iterative dialogue between the analyst and the visual analytics system as follows (Figure 1, 2):

1. **System – Show and predict:** The system shows the current state of the data as well as predictions for the future at the selected position in time using visual representations.
2. **Analyst – View and select:** The analyst views the visualized current state and predictions (and their likelihoods), and can make one of three choices about future change:
   (a) *Manual:* The analyst has knowledge or intuition that conflicts with the predictions proposed by the system, so chooses an entirely different projection for the future than suggested by the system.
   (b) *Hybrid:* The analyst selects one of the proposed future predictions suggested by the system.
   (c) *Automatic:* The analyst accepts the system's choice as the most likely prediction.
3. **System – Accept and update:** The system uses the analyst's prediction as a "what-if" question (what if the chosen prediction is true) to create new predictions (conditional predictions) for the time series, essentially *forking time*. Return to Step 2 (or Step 1 if no predictions exist).

Step 2 is the analyst's input point in this dialogue. Beyond making a choice about the next prediction action, this is also where the analysts can change the time step (allowing for a different prediction duration), move to a different point in time (essentially abandoning the current line of prediction), and go back to previous actions (equivalent to forking the prediction). This is also where an analyst can decide to stop entirely, presumably because the new prediction state provides sufficient information to make a decision.

An actual implementation of TimeFork in a visual analytics system needs to be instantiated with specific computational models, visual representations, and a prediction interface that are all specific to the domain and dataset.

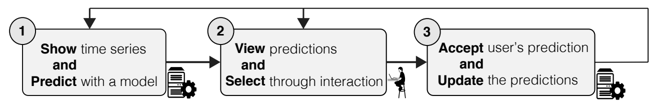

Figure 2. The three-step TimeFork workflow maintains a dialogue between the analyst and the computer for visual prediction.

TIMEFORK FOR STOCK DATA

To showcase the TimeFork technique, we developed a stock market analytics tool, STOCKFORK, that uses the TimeFork technique to support visual prediction of the stock prices. Our dataset consisted of end-of-day stock price (*adjusted closing*) and trade volume for stocks from the technology and consumer goods sectors (data from 2012 to 2015). TimeFork's interactive prediction workflow strongly relies on user understanding of the time-series data and also the factors influencing it. Therefore the StockFork interface is designed to visualize the stock prices and trade volume, which is the primary quantitative information stock traders and analysts follow, as well as aid analysis across stocks through derived attributes (e.g., cross-correlation) [52]. Besides this, external information from the domain is also shown to aid prediction. Our visualization design makes it easy to plug in the TimeFork technique, thus, exemplifying how it can be supported in other time-series visual analytics tools [6, 22, 52, 53].

StockFork: Visual and Interaction Design

StockFork (Figure 3) contains an overview+detail layout following Shneiderman's guideline [42] for visual exploration—overview first, zoom & filter, details on demand. The overview captures overall patterns in the stock market data, and the detail views visualize selected time periods.

Overview representation. The overview in StockFork captures the overall patterns for multiple stocks of interest, selected from a stock list available in the interface. While there are many design choices for showing multiple time series in a single view (as outlined by Javed et al. [25]), our overview design contains horizon graphs [20] stacked vertically visualizing stock prices in a space-efficient layout (seen in Figure 3). Horizon graphs are effective for discrimination (is point A higher than point B?) [25] and thus enable the users quickly understand the past trends. Other design choices for horizon graphs include offset and mirror representations with different band count and height, which have their own effects on graphical perception [20].

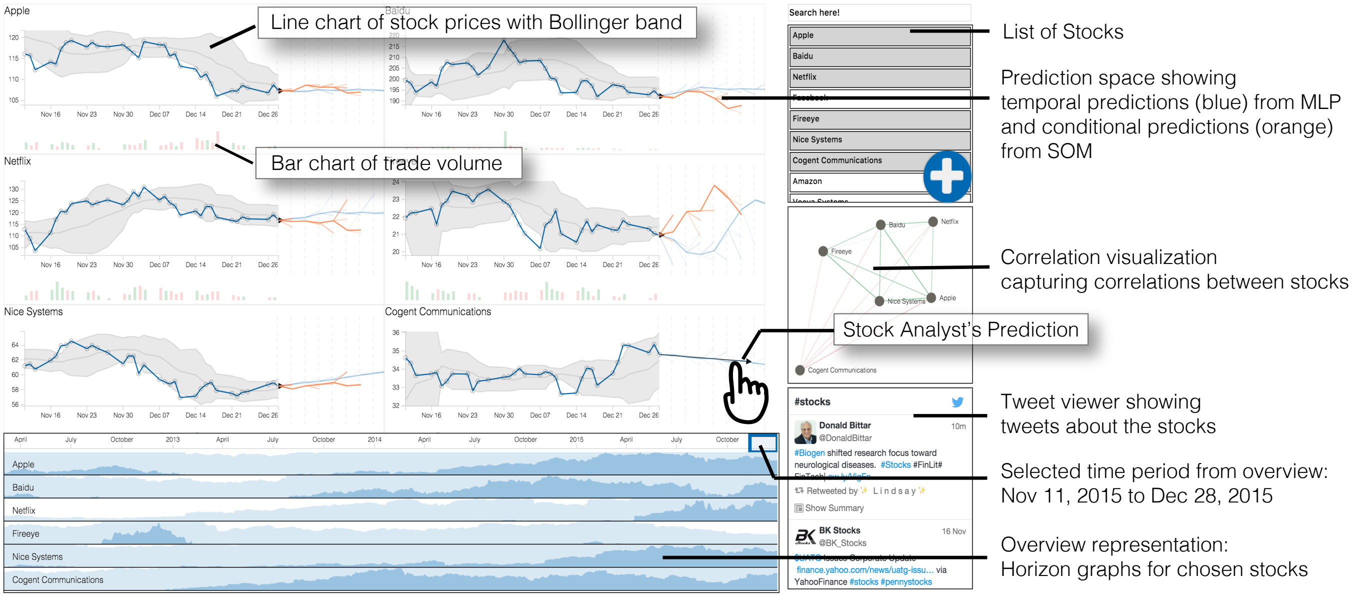

Figure 3. StockFork presents line charts of stock prices along with other visualizations and prediction space to support TimeFork for visual prediction.

The overview also supports selection of a time period on the time axis to show specific data in the detail views. This helps the analyst view different time scales (e.g., past week vs. month), which is common in stock trading, and also test TimeFork on a past time period against ground truth data.

Detail representations. The detail views capture visual information that can aid in visual prediction using the TimeFork technique. StockFork has three detail views:

- *Small multiple charts:* These charts visualize the stock price and trade information. Our design includes line charts for stock prices and bar charts for trade volume (one per stock), as they are common representations for time-series data. The line charts also show Bollinger bands [4] capturing moving averages and moving standard deviations, which indicate bullish/bearish trends in stock prices. Furthermore, the bar charts use color (green, red) to distinguish trades leading to rise/fall of stock price. These features are commonly seen in stock trading tools [41]. More design choices do exist including circular layouts [53] and complex representations such as Candlestick plots [36].

- *Correlation visualization:* Exploratory analysis of time series also requires supporting elaborate tasks such as visualizing derived values and identifying correlations [52]. In StockFork, we chose to visualize correlations between stocks to understand which stocks show similar trends (a task pursued by analysts working with multiple stocks). We use a visual design inspired by Moere et al. [35] for this purpose—a node-link diagram using position of nodes (stocks) to encode Pearson correlation (r) between stock prices in the selected time period. The links in the diagram are colored (green, red) based on the sign of r $(+, -)$, with opacity capturing $|r|$. Stocks that are positively correlated are placed closer to each other (using a force-directed layout), thus, implying they might follow similar price trends.

- *Tweet viewer:* Beyond the stock market dataset, external information from public opinions and company reports are often used as qualitative information sources for prediction. StockFork streams tweets related to the stocks using a Twitter widget to represent such information.

Enabling TimeFork's Interactive Prediction Workflow

To support the TimeFork workflow, the line charts are attached with a prediction space showing predictions from the computational models and supporting user interaction.

Step 1: Show and Predict (System)

In this step, temporal predictions are generated by a computational model for the stocks based on the past values in the selected time period. The model driving this step is a multi-layer perceptron (one MLP model per stock), a feed-forward neural network model, trained on the stock price data from our dataset for each stock. Each neural network model has five layers: input layer of 6 neurons, hidden layers of sizes $(50, 60, 70)$, and an output layer of one neuron, with Sigmoid activation function. The input to each model is the past six relative price changes (ratio of price change to previous stock price), and the output is the relative future change (decoded using the current stock price). The models are trained on these $(input, output)$ pairs from the historical data using the standard backpropagation algorithm. Trained models showed more than 90% accuracy with the training dataset.

Apart from these predictions directly obtained from the past values, alternative predictions of lower confidence values are also created at the same time by changing one or more input elements (by a 10% fixed margin) and recording the model's output. The confidence value ($c \in [0, 1]$) of an alternative prediction was scaled inversely by the number of changes made to the input. Also, the choice of the margin was based on a manual verification of different margins (5%, 10%, and 20%) to find the one that reinforces top (best) predictions from the

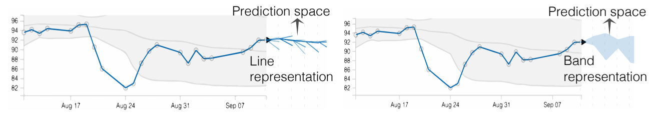

Figure 4. Two different forms of visualized prediction information: lines showing individual predictions of different confidences mapped to opacity and bands showing the range of possibilities.

training data with less-variant alternatives. These alternative predictions can therefore give a sense of variation of the predictions based on the change in the input pattern, and provide an awareness of the model's confidence in the predictions. Although predictions are for a single day each, longer time periods (e.g., a week) can be predicted using individual predictions for each day as ground truth to predict the next. We refer to this approach as *prediction chaining*. Note that prediction chaining can lead to a tree structure of predictions, however, we have chosen to work with the main trunk (high confidence predictions) and its alternatives rather than all branches to minimize visual clutter when rendered (leading to the structure in Figure 4).

Step 2: View and Select (Stock Analyst)

The predictions from Step 1 can be visualized individually or through aggregations showing a range [7]. Figure 4 shows both prediction representations from the StockFork interface. The line representation is a direct presentation of the predictions, with each line showing the change, and the opacity capturing the prediction's confidence. For the band representation, the individual predictions are processed to compute a weighted mean (μ) and standard deviation (σ) for each day in future to visualize a band ($\mu \pm 2\sigma$). The choice of the prediction representation is based on the time-series visualization itself: while line and band representations fit well with a line chart, other visualizations may require a different prediction representation that aligns better with the overall design.

User interactions with the prediction space are then enabled. Analysts can proceed with the gained visual knowledge from the detail views and the visualized model predictions in three ways: (1) neglect the visualized predictions altogether and make their own prediction (manual prediction); (2) select one of the proposed predictions that best suites their understanding (hybrid prediction); and (3) accept the most likely prediction provided by the computational model (automatic prediction). Analysts make one of these interactions directly on the prediction space by choosing a direction and a time period of predicted movement (by dragging an arrow on the prediction space signifying movement using mouse or touch input). This interaction is then fed into the system in the next step.

Step 3: Accept and Update (System)

The Accept and Update step is the most crucial part of the TimeFork technique to create and maintain an interactive dialogue with the system. Analyst predictions from the previous step reflects their understanding of what might happen to a stock. Given that stocks patterns may co-occur—e.g., Apple releasing an iPhone can effect Samsung's stock—computational models in this stage consider the predictions for some stocks and generate new (conditional) predictions for others to reflect the analyst's understanding. Therefore, models for this step must cluster or classify similar co-occurrence patterns from the historical data (e.g., Apple and Tesla increased together by 5% each over 100 times in the past), and enable quick lookup of similar patterns (if Apple increases, what will happen to Tesla?). For this purpose, we used a self-organizing map (SOM) [29] of 625 neurons and trained it on the co-occurrences from the dataset (e.g., Apple and Tesla have relative price changes of 0.05 and 0.03 on a day). Each neuron after training captures a co-occurrence pattern across all stocks, and neurons are clustered by the Euclidean distance between these patterns. Upon receiving the user interaction, which is divided into day-level predictions through the *closest fit* in the prediction space, the SOM is searched for the cluster of neurons with similar trends for the stocks predicted by the user and conditional predictions are obtained from it. Prediction confidences are mapped to the Euclidean distance of the cluster neurons from the pattern.

Similar to the predictions from Step 1, conditional predictions are generated for each time step and are chained to cover longer time periods. The conditional predictions can be integrated with the visualized temporal predictions from the previous step, or presented as a new band/line. This step thus enables the computer to communicate its own assessment for other stocks based on the analyst's prediction, enabling visual exploration of predictions and maintaining a continuous dialogue (Figure 5), which ends with the analyst gaining enough information to make an investment decision.

Other interactions. StockFork also supports storing the user predictions from Step 2 (*save*) and reverting to a previous state of the prediction space (*undo*) in the interface.

StockFork: Implementation

Our StockFork implementation uses a server-client architecture and is available as open source[1], along with sample stock market data and prediction models. The client interface was developed with web technologies—HTML5, JavaScript (JS), and CSS3—with D3 visualization framework [5] for the overview and detail views. StockFork's server uses Node.js to (1) collect stock market data using Yahoo Finance API (through Historic library [11]), and (2) train the models to enable TimeFork's interactive prediction workflow on the client. To refresh, two types of models were used: multilayer perceptrons (MLP) in Step 1 for temporal predictions and a self-organizing map (SOM) in Step 3 for conditional predictions.

Temporal Predictions

The multilayer perceptrons (one per stock) were implemented with the Brain library [1]. This model is trained on relative price changes, as described earlier, and predicts the future changes thus providing temporal predictions. The training procedure happens ahead of time and each trained MLP is converted into a serialized format (a JSON file) capturing all neuron weights. The pre-trained model can be recreated on the client using this file and the Brain JS API. This is carried out by a StockFork wrapper, which also accesses the pre-trained model to generate the temporal predictions.

[1] https://github.com/karthikbadam/TimeFork

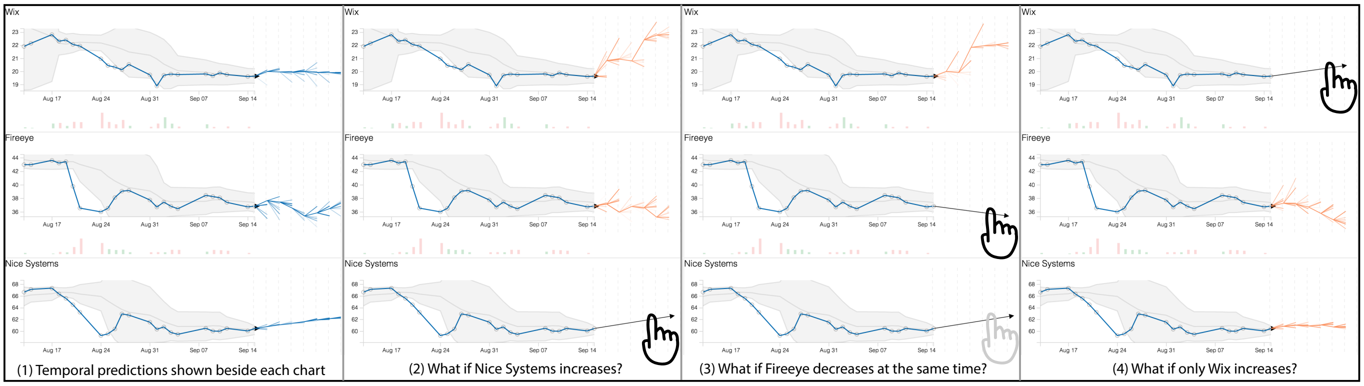

Figure 5. A typical dialogue in StockFork: the analyst interacts and the computer updates the predictions in the visualizations for further exploration.

Conditional Predictions

The self-organizing map was implemented with the ML-SOM library [51]. The SOM is trained on co-occurrence patterns of relative changes ahead of time. It is converted into a JSON file consisting of the weights and passed to the client where the model is recreated. A StockFork wrapper takes input (i.e., user's prediction for one or more stocks), accesses this pre-trained model on the client to generate conditional predictions, thus, answering the "what-if" questions.

New models for temporal and conditional predictions can be easily added by developing client-side JS wrappers that take the corresponding input (e.g., past relative price changes for temporal prediction) and return the model predictions.

USER STUDY

The TimeFork technique creates a human-computer dialogue by visualizing predictions, allowing the user to predict through manual, hybrid, and automatic predictions, and updating the predictions for further exploration. While manual and automatic prediction reflect traditional approaches in stock market (and time series in general), the presence of all three choices in TimeFork helps the analysts account for their own knowledge and understanding to come to a shared set of predictions with the computer. Therefore, the focus of our user study was not only to understand if TimeFork leads to better predictions, but also when/why it does so and how it changes the analyst's approach towards prediction and the entailing decisions. For this reason, we used simulated stock trading tasks where the participants can read multiple visualizations of stocks and information about the companies, gain an understanding, and engage in the dialogue.

Participants

We recruited 13 participants (5 female, 8 male) from the general population (including students, faculty, and staff) within our university campus. The participants were between 18 and 50 years of age. They were paid $10 for participation. All participants self-reported as proficient computer users with 6+ years of experience. Furthermore, 11 participants had previously used visualizations for data analysis and 7 of them had experience following stock markets; however, it was limited to either investing through a trader or novice-level experience in technical analysis of stock markets.

Dataset and Task

We picked stock market analysis due to the natural interest that many people have in stock trading and the potential opportunity to gamify the task by providing the participants with virtual money to invest. This gamification can help incentivize our participants to engage in TimeFork's workflow and make informed decisions to succeed at the tasks, while still maintaining the ecological validity of our experiment. The participants were given $100,000 in virtual money and asked to invest in three stocks: Apple, Facebook, and Tesla, on a simplified StockFork interface that shows line charts for stock prices, summarized earnings reports (1-2 paragraphs per company), and the prediction information from the models (when TimeFork is enabled). Quarterly earnings reports were used instead of Twitter, as they are publicly available and more commonly used by traders. These reports detailed the earnings and profits made by each company (and its products) compared to previous quarters, as well as investments in new ventures. The stocks were picked after a discussion with a stock trader as a representative set of a popular long-term player (Apple), a growing stock (Facebook), and a volatile stock (Tesla). The selection criteria for these stocks is not related to TimeFork but rather a representative of a stock portfolio with stocks of different risk values.

The task consisted of asking participants to predict the behavior of each stock and make a decision on how much to invest on them. Following this, they had to click an *evaluate* button on the interface that invests the said amount over the chosen time period and calculates their final earnings from this trade.

Experiment Factors

The prediction technique is a factor (T) influencing participant performance. Since fully automated prediction does not require a user, our experiment consisted of two conditions:

(A) TIMEFORK PREDICTION: In this condition, the participants are shown the simplified StockFork interface with TimeFork enabled (Figure 6). The prediction information is shown and the system reacts to user interaction.

(B) MANUAL PREDICTION: Here, the interface is similar to the previous condition except it lacks the prediction information due to the absence of TimeFork.

Experimental Design and Procedure

During the study, each participant predicted the stocks in both Technique (T) conditions. To counter learning effects, we picked two sets of data for using the techniques (one per technique) and introduced a factor D that represents the stock data shown: (***D*1**) data after the second quarter reports in July 2014 (Jul-Aug 2014), and (***D*2**) data after the fourth quarter reports of 2014 released in Jan 2015 (Jan-Mar 2015).[2] The participants were asked to invest their earnings on an assigned interface (technique + data combination) through four time steps (repetitions) within the assigned data. They were shown stock price data over 3-4 weeks and were asked to make predictions spanning between 1 and 20 days into the future based on their assessment in each time step. The participants were also shown the performance of their prediction in the previous time step (in the assigned D), to revise their strategies if needed. Overall, the participant's goal in an interface is to maximize their overall profit by investing their current earnings in the three stocks at each time step. The order of the technique (T) and data (D) factors was counterbalanced.

An experiment session began with the participant arriving, and reading and signing a consent form. The participants then completed a demographic survey. Following this, they were shown how to use the interface for the assigned first condition, including how to read the line charts, how to explore the prediction space, and how to invest on the stocks. The participants trained using the interface (with a training dataset), during which they were encouraged to ask questions, until they were comfortable. They then finished the tasks, followed by completing a Likert scale survey rating the efficiency, ease of use, and enjoyability of the technique. After the first condition, the participants continued to do the same with the second condition. Each session lasted between 45 and 60 min.

We recorded the profit/loss made during the tasks, along with the investment information. The participant interactions were recorded for further analysis. The participants were also encouraged to "think aloud" and announce their understanding of the stock behavior. The sessions were followed up with a post-session interview to examine their strategies to understand the time-series data, deal with the reactions of the computer to their interaction, and decide on the investment.

Study Design Rationale

Here we discuss our study design decisions and rationale.

- *Real stock market data.* TimeFork strongly relies not only on the analyst's understanding of the visualized data but also on user's domain knowledge that is not captured by the models. We therefore used stock market data along with earnings reports. Furthermore, we used real stock names to maintain the practicality (also, the reports contain other identifiable information). Counterbalancing the conditions helps reduce any participant bias linked to this decision.

- *Novice participants.* We recruited from the general university population rather than expert traders to ensure that the prior knowledge of the stock trends is minimal. Furthermore, we envision TimeFork to be accessible not just to experts but also novice analysts who are computer literate. Our participants were requested to make decisions based on the understanding gained during the experiment.

- *Simplified study interface.* We chose to limit the study to a simplified interface of StockFork to focus on the mechanism being studied here—prediction of time series. None of the removed views were necessary for the study.

- *Multiple datasets.* Two separate sets of stock market data are chosen to work with the two techniques (since exposure to a data once would influence the performance if used again). Within each set, the prediction task is repeated four times using four time steps. This procedure reduces the effects of randomly guessing the future.

[2]The prediction models were trained only with the stock market data from July 2012 to February 2014 for the study.

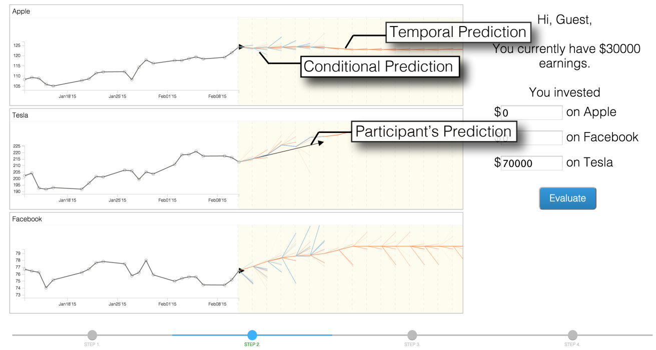

Figure 6. StockFork interface used in the user study (with TimeFork).

RESULTS AND OBSERVATIONS

Below we report the results from the statistical analysis and the qualitative analysis, as well as the participant feedback.

Quantitative Performance

Similar to Green et al. [15], we used a (linear) *mixed effects model* for our analysis of profit to avoid the fixed-effect fallacy [10]. We modeled participant and data D as random effects with a random intercept term for participant and random slope+intercept for data. Technique T was our fixed effect. Overall, we found no significant main effect ($\chi^2(1, N = 26) = .809$, $p = .37$) of technique—TimeFork (mean = \$14,907.18, s.d. = \$11,978.27) was similar to Manual prediction (mean = \$7,474.32, s.d. = \$4,944.23). However, we found differences in the coefficients (intercept, slope) for the effect of TimeFork in data $D1$ and $D2$ (discussed in the supplementary material).

Performance Differences in Data Conditions

We then analyzed the participant performance individually for both data conditions (Figure 7)—stock data from Jul-Aug 2014 ($D1$) and Jan-Mar 2015 ($D2$)—using the independent-samples T-test (with Bonferroni correction). In 2014 data ($D1$), there was a significant performance difference between TimeFork and Manual ($t(11) = -3.34$, $p = .013$). Participants using TimeFork (mean = \$23,467.20, s.d. = \$9,061.30) outperformed Manual prediction (mean = \$9,831.10, s.d. = \$4,454.80) for tasks with data $D1$. In 2015 data ($D2$), there was no significant difference between the two techniques.

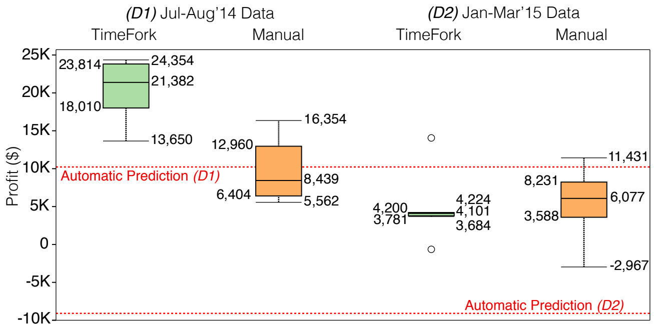

Figure 7. Box plot capturing the distributions—median (center line in each box), interquartile range (box), extrema (whiskers), and outliers (circles)—of performance using TimeFork and Manual prediction with Data $D1$ and $D2$. Performance of automatic prediction is highlighted.

Performance of Automatic Prediction

We analyzed the capabilities of the computational models by calculating the performance if the model's best predictions were followed (automatic prediction). The investment strategy here consists of following the high confidence predictions till the day when their direction changes from a rise to fall in stock price. The slopes of these rising prediction trends are used to split the earnings proportionally across stocks. No investment is made on stocks whose best predictions show an immediate fall in stock price. For example, if Apple increases for the next 3 days by 5% and Tesla increases for the next 10 days by 5% (while Facebook decreases), more money is invested in Apple, than Tesla, due to the promise of a quick turnover. Based on this, we found that our models generate \$10,225.60 profit for the initial \$100,000 virtual money provided for 2014 data ($D1$). For 2015 ($D2$), the model lost \$9,094.80 showing that the model failed to predict the stocks.

Investment Differences

Since the participants were free to invest any fraction of the total earnings, we were interested in observing the differences in investment across stocks for each data condition. Similar analyses as before with T-tests revealed significant differences in investment between TimeFork and Manual in data $D1$ for Apple ($t(38) = 3.46$, $p = .008$) and Tesla ($t(45) = -3.65$, $p = .004$). For Tesla, TimeFork (mean = \$47,389, s.d. = \$28,776.20) had higher investments than Manual (mean = \$20,420.40, s.d. = \$19,699.70). In contrast for Apple, Manual prediction (mean = \$53,008.60, s.d. = \$24,640.80) had higher investments than TimeFork (mean = \$29,929.50, s.d. = \$16,226.40). In $D2$, there were no significant differences in investment between the two techniques.

Qualitative Analysis

To understand the results from statistical analysis and the extent to which participants explored StockFork, we present our observations from the study in the following five categories:

Visual Understanding of Time Series through StockFork

All tasks began with the participants observing the stock price trend in the line charts, reading the earnings reports, the performance of their previous prediction, and sometimes, directly exploring the predictions shown by the system (when TimeFork is enabled). In TimeFork condition, eight participants (P1, P2, P5-P8, P10, P13) started their tasks by looking at the reports, line charts, and their past predictions. The rest started by first reading the predictions shown in the TimeFork condition. In Manual prediction, all participants started by reading the reports and the line charts along with their previous predictions. While reading the charts, the participants looked for (1) a dominant trend in the past 7-10 days to see if it's going up or down (P1-P13), (2) the range of the stock price and the variance seen in the line chart (is stock price fluctuating and by how much—P4, P8, P12), and (3) the number of times the stock price changed direction (P3 in Manual).

Dialogue through TimeFork

In StockFork, the information sources described earlier often give a conflicting impression of what might happen. For example, Apple might be doing well on the earnings report in terms of its product sales and profits, but the current trend in its line chart might be decreasing. TimeFork's predictions added within the interface can complicate this further as it might not be fully agreeing with other visualized information. The user interactions in the TimeFork workflow—manual, hybrid, and automatic predictions—are therefore useful to come to a common understanding through an interactive dialogue. Overall, there were three levels of dialogue observed:

- **Minimal Dialogue:** Participant (P9), in two of the four task repetitions, started his prediction process by selecting the best temporal prediction path given by the system for a stock (automatic prediction), to see that the predictions for others are fluctuating (i.e., the variations in the lower confidence conditional predictions are high). He resolved this conflict by accepting the best temporal prediction path for each stock, and neglecting other visualized information.
- **Moderate Dialogue:** Eleven participants (except P3 and P11) preferred to start exploring the computer's predictions by either making a manual prediction (based on earnings report and chart trend) or choosing the system's best temporal prediction. This led to conditional predictions for other stocks, which are explored through 3-4 interactions to either make a manual prediction for these stocks, or find a prediction pattern that they most agree with based on the earnings reports and the chart trends. In the latter case, participants mostly chose a lower confidence computer prediction (hybrid). Participant P1 said that he wanted the computer to "*weigh in*" on what might happen, but he did not necessarily follow the best suggestion that was provided. The performance of the participants engaging in moderate dialogue was varied depending how they approached the time period of prediction and the investment.
- **Extended Dialogue:** Five participants (P3, P4, P6, P8, P11) indulged in an extended dialogue, with P3 and P11 using it for all task repetitions. In this case, the participants would interact through all three types of predictions—manual, hybrid, and automatic—to find the conditional prediction trend that best aligns with the reports and the charts, and shows least variance in the prediction space. These participants also adjusted the time period for their predictions during this process. They would typically go

through more than six interactions (with P3 making up to 20 interactions), to arrive at their answer. It is worth noting that participants P3, P4, P6, P11 used TimeFork with data $D1$ and performed better than both automatic and manual prediction conditions, with P3 making the highest profit.

Strategies: Direction of Prediction

In TimeFork condition, the direction of the user's prediction was directly influenced by their dialogue with the system. P9, who preferred minimal dialogue, used the direction of the system's best temporal prediction. Participants in moderate dialogue, had multiple strategies for direction: (1) when they encounter conflicting predictions (predictions that do not reflect the reports, chart trends, or too different to each other), made a manual or hybrid prediction based on the earnings reports (P1, P4, P7, P12, P13) and the chart trends (P2, P4, P5, P9, P12, P13); and (2) the more popular strategy, was to come up with a manual prediction that is between the temporal predictions and the conditional predictions for the stocks, in case of conflicts. The second strategy was observed at least in one repetition for ten participants (all except P3, P9, P11). Between P3 and P11, who had an extended dialogue, the direction was based on the best conditional predictions from visual exploration. Finally, in Manual prediction, participants either extrapolated the trend in the past 10 days (all participants) or selected an increase/decrease direction based on the earnings reports (all participants except P3). P8 also incorporated his previous prediction performance in his strategy.

Strategies: Time Period of Prediction

In TimeFork, P9 followed the system's best predictions to decide the time period. Participants in a moderate dialogue used different strategies based on their exploration: (1) for hybrid predictions, the time period was based on the computer's (lower-confidence) predictions (11 participants); and (2) for manual predictions it was a selection of a short time period (2-5 days—P1, P2, P5, P7, P10, P13). Participants in the extended dialogue made this decision based on the dialogue. In Manual, the participants chose a short (2-5 days—P2, P8, P10) or long time period (6-15 days—P3, P4, P11, P12, P13) without a significant reason. Some participants (P5, P6, P7, P9) chose a time period based on the trend in the chart.

Strategies: Investment

The investment in both techniques was mainly based on the participant's predictions—for twelve participants (P2-P13) in TimeFork and eleven (P1, P4, P6-P13) in Manual prediction. In TimeFork, participants (P1, P5, P7, P10) also used the variance of the computer's predictions and the chart trends for their investments. In Manual condition, participants P2, P5, P10 used the reports as a measure of "confidence" for investment, and participant P4 invested more in charts with less variance. After reviewing his previous predictions, P3 invested all his earnings on a single stock that he concluded was increasing the most based on his visual exploration.

Subjective Ratings and Feedback

After each session, the participants rated the techniques on three metrics: efficiency, ease of use, and enjoyability, on a Likert scale ranging from 1 (e.g., strongly disagree) to 5 (e.g., strongly agree). We analyzed these ratings with Wilcoxon Signed Rank tests and found no significant differences between the two techniques. However, most participants (P2-P4, P8, P10-P13) mentioned that Manual prediction required more guess work. They also felt that it took them more time to get used to the TimeFork's workflow. P2 stated that TimeFork gave her "*someone else's point of view*," while P3 and P4 expressed that the conditional and lower-confidence predictions showed how sensitive other stocks to a particular stock.

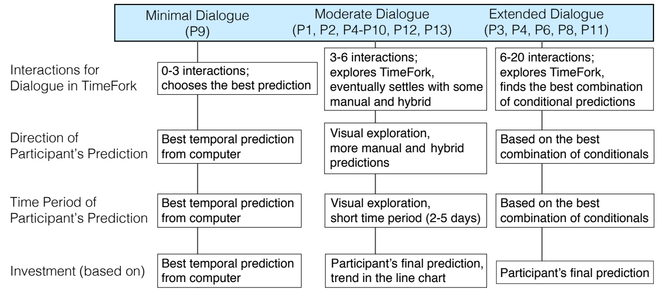

Figure 8. Three forms of dialogue and corresponding usage patterns of the TimeFork technique observed through a qualitative analysis.

DESIGN IMPLICATIONS

Our StockFork application instantiates TimeFork using neural network models to provide temporal and conditional predictions that contribute to the dialogue with the analyst. However, this also severely influences the visual prediction. In our experiment setup, the computational models performed better for 2014 data ($D1$) compared to 2015 ($D2$), and this led to participants performing better with TimeFork for $D1$, compared to no difference between TimeFork and Manual prediction in $D2$. This suggests that having a dialogue with an intelligent partner (accurate model) affects the analyst's ability for the better—it led to significantly higher profits in TimeFork outperforming both automated and manual prediction. On the other hand, TimeFork's current design did not manage to overcome the model weaknesses (inaccuracies in prediction). Participant performance was not affected by the presence of a weak model in $D2$, but it changed the level of dialogue with the computer. More work is required not only to measure the effects of weak models, but also to create methods for visualizing the weakness of computational models.

As expected, most participants (twelve) always engaged in the dialogue (moderate or extended) with the computer; however, the amount of dialogue affected the visual predictions and the decisions following it. This explains the investment differences—the presence of a good dialogue in $D1$ guided the distribution among stocks. Participants in extended dialogue explored a range of "what-if" scenarios that led them to shared predictions with the system. They were observed to be consistent in their decisions: direction and time period of prediction (directly obtained from this exploration) and investment (mapped to their predictions), and also performed better than the automatic, manual prediction, and average performance in their condition. However, TimeFork currently only succeeded in motivating five out of the thirteen participants to explore an extended dialogue. Also, participants (4 of 5) using the strong (accurate) model in $D1$ mostly engaged in this dialogue. Participants preferred to override the computer to some extent and make manual predictions rather than engage in a fully extended dialogue. For this reason, we recognize a potential weakness of TimeFork in lacking an overview representation of "what-if" scenarios to engage the user.

The decisions entailing predictions are often more important than the predictions themselves. For example, an energy provider analyzing the consumption of gas and electricity may visually predict that the gas consumption might increase while electricity decreases. However, they would be more interested in understanding how to deal with this and distribute their energy resources. In our study, participants mostly mapped their investments directly to their predictions, but, four of them also considered the confidence of the computer itself into account ("if the computer's predictions are too varied, maybe I should not invest too much"). This effect may lead to positive or negative results as it connects to why the computer's predictions are varied and also the user's interaction itself. More work is required to understand ways to mitigate this effect either by decoupling prediction and decision making, or by embedding solutions that propose possible answers to these decisions based on the user interactions.

Limitations. Our study design decisions have unique influences on the outcome. Firstly, the two datasets (D) follow the release of the quarterly earnings reports to ensure that there is an external factor (the earnings reports) influencing the stocks, and they are separated by six months so that the trends are not influencing each other a lot. However, this choice has led to a performance difference in automatic prediction mainly since data $D2$ had more instances of stock prices falling. Also, earnings reports may reflect only a subset of all external factors affecting the stock prices. Secondly, investing in three stocks at once in each task meant that the profits made on one stock influenced others (therefore, overall performance was analyzed). A more controlled study is needed to understand the effects of TimeFork on individual stocks and specific trends. Finally, real stock names may have impacted the performance of our novice participants; however, this is important as knowledge of external factors facilitates the users better express their opinions in the dialogue with the computer system to find the best predictions.

APPLICATION SCENARIOS

While we demonstrated our TimeFork technique with stock market data, we believe that the technique is applicable to other scenarios as well. In this section, we provide two short application scenarios inspired by real-world challenges.

Household Resource Consumption

Consider an energy provider company interested in understanding resource consumption (electricity, water, and gas) in a household neighborhood. The company wants to manage their resource reserves using years of usage data, along with data from temperature, humidity, and moisture sensors in the households and the neighborhood. In the presence of such fine-grained, high-volume, and high-velocity streaming data, the company is trying to understand which household will have an increased consumption of electricity. Since there are often relationships between the variables in that data, interactive visual prediction through TimeFork works well. The company's analysts would look at the visualized time series from the neighborhood and the households, and start by predicting what might happen in the neighborhood (based on TimeFork or their own knowledge). They would then see updated predictions for the electricity, water, and gas consumptions for each individual household. They could further filter these predictions by choosing a trend for the temperature within these households. They could choose not to agree with the system at this stage, and make a different prediction based on temperature and humidity, to find the predictions to be more conclusive. Upon finding the households with predicted increase in electricity consumption, they could use a similar visual exploration to estimate the gas and water consumptions for proper resource allocation or maintenance.

Movie Earnings

Let us consider a more casual example in the entertainment industry. Hundreds of movies are released per year just in the United States. Often the earnings of the movies over time depend on the cast, genre, and also other movies releasing in the same time period. To figure out the ideal release date for a new movie [38], an analyst can access the historical patterns of movie earnings and releases on a TimeFork-enabled VA interface. She would then try to find the seasonal effects on the new movie of a particular genre (say comedy) and cast, by expressing that there will be a comedy movie in April-June season, to see the predictions for earnings. Following this, she can also postulate that there might be three other movies releasing in the same week with different genres, to see their effect on the earnings of the original movie. She could further change the season, and also query based on the cast, to find the ideal time period for the movie release.

CONCLUSION AND FUTURE WORK

We have presented the TimeFork method for interactive visual prediction based on combining predictions from computational forecasting models with analyst input in an interactive interface. To showcase the idea, we have further applied it to stock market data. Our user study showed significantly better performance in the presence of an efficient dialogue when using TimeFork compared to manual (in data $D1$) and automatic prediction (in data $D1$ and $D2$). In other words, our work shows that involving both humans as well as computational models in the sensemaking loop is a useful approach.

The TimeFork method represents a canonical example of visual analytics [46] and we are eager to continue exploring such explicit human-computer collaborations. However, one of the biggest challenges in visual analytics design is to identify the inflection points where the analyst can provide the most useful feedback to the system. The TimeFork method provides one such design, and we see much potential for applying the method to other datasets and domains.

ACKNOWLEDGMENTS

We thank the anonymous reviewers, Catherine Plaisant, Cody Buntain, Matthias Nielsen, and Senthil Chandrasegaran for their feedback that substantially improved our manuscript.

REFERENCES

1. Heather Arthur. 2013. Brain Node.js Library. (2013). https://www.npmjs.com/package/brain, accessed December 2015.
2. Guilherme A. Barreto. 2007. Time series prediction with the self-organizing map: A review. In *Perspectives of Neural-Symbolic Integration*. Springer, 135–158.
3. Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2, 1 (2011), 1–8.
4. John Bollinger. 2001. Bollinger On Bollinger Band. (2001).
5. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309.
6. Paolo Buono, Aleks Aris, Catherine Plaisant, Amir Khella, and Ben Shneiderman. 2005. Interactive pattern search in time series. In *Proceedings of SPIE Electronic Imaging*. 175–186.
7. Paolo Buono, Catherine Plaisant, Adalberto Simeone, Aleks Aris, Ben Shneiderman, Galit Shmueli, and Wolfgang Jank. 2007. Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting. In *Proceedings of the International Conference on Information Visualization*. 191–196.
8. Philip K. Chan and Matthew V. Mahoney. 2005. Modeling multiple time series for anomaly detection. In *Proceedings of the IEEE International Conference on Data Mining*. 90–97.
9. Bill Chiu, Eamonn Keogh, and Stefano Lonardi. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*. 493–498.
10. Herbert H Clark. 1973. The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of verbal learning and verbal behavior* 12, 4 (1973), 335–359.
11. Nick Desaulniers. 2013. Historic Node.js Library. (2013). https://www.npmjs.com/package/historic, accessed December 2015.
12. Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for visual text analytics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 473–482.
13. Tak-chung Fu. 2011. A review on time series data mining. *Engineering Applications of Artificial Intelligence* 24, 1 (2011), 164–181.
14. Tak-chung Fu, Fu-lai Chung, Ka-yan Kwok, and Chak-man Ng. 2008. Stock time series visualization based on data point importance. *Engineering Applications of Artificial Intelligence* 21, 8 (2008), 1217–1232.
15. Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 439–448.
16. Ming Hao, Halldór Janetzko, Ratnesh Sharma, Umeshwar Dayal, Daniel Keim, and Malu Castellanos. 2009. Poster: visual prediction of time series. In *Poster Compendium of the IEEE Symposium on Visual Analytics Science and Technology*.
17. Ming C. Hao, Umeshwar Dayal, Daniel A. Keim, and Tobias Schreck. 2007. Multi-resolution techniques for visual exploration of large time-series data. In *Proceedings of the Joint Eurographics / IEEE VGTC Conference on Visualization*. 27–34.
18. Ming C. Hao, Halldor Janetzko, Sebastian Mittelstädt, Water Hill, Umeshwar Dayal, Daniel A. Keim, Manish Marwah, and Ratnesh K. Sharma. 2011. A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series. In *Computer Graphics Forum*, Vol. 30. 691–700.
19. Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. 2010. A tour through the visualization zoo. *Communications of the ACM* 53, 6 (2010), 59–67.
20. Jeffrey Heer, Nicholas Kong, and Maneesh Agrawala. 2009. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 1303–1312.
21. Magnus Lie Hetland and Pål Sætrom. 2005. Evolutionary rule mining in time series databases. *Machine Learning* 58, 2-3 (2005), 107–125.
22. Harry Hochheiser and Ben Shneiderman. 2004. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization* 3, 1 (2004), 1–18.
23. Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 159–166.
24. Eric J. Horvitz. 2007. Reflections on challenges and promises of mixed-initiative interaction. *AI Magazine* 28, 2 (2007), 3.
25. Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. 2010. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics* 16, 6 (2010), 927–934.
26. Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern Recognition* 44, 9 (2011), 2231–2240.
27. Ashish Kapoor, Bongshin Lee, Desney S. Tan, and Eric Horvitz. 2012. Performance and Preferences: Interactive Refinement of Machine Learning Procedures. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

28. Daniel Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. 2006. Challenges in visual data analysis. In *Proceedings of the International Conference on Information Visualization*. 9–16.

29. Teuvo Kohonen. 2001. *Self-Organizing Maps*. Springer.

30. T. Warren Liao. 2005. Clustering of time series data—a survey. *Pattern Recognition* 38, 11 (2005), 1857–1874.

31. Jessica Lin, Eamonn Keogh, and Stefano Lonardi. 2005. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization* 4, 2 (2005), 61–82.

32. Andrew C. Lorenc. 1986. Analysis methods for numerical weather prediction. *Royal Meteorological Society, Quarterly Journal* 112 (1986), 1177–1194.

33. Junshui Ma and Simon Perkins. 2003. Online novelty detection on temporal sequences. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*. 613–618.

34. Abish Malik, Ross Maciejewski, Yun Jang, Whitney Huang, Niklas Elmqvist, and David Ebert. 2012. A Correlative Analysis Process in a Visual Analytics Environment. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. 33–42.

35. Andrew Vande Moere. 2004. Time-varying data visualization using information flocking boids. In *Proceedings of the IEEE Symposium on Information Visualization*. 97–104.

36. Steve Nison. 2001. *Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East*. Penguin.

37. Ajoy K. Palit and Dobrivoje Popovic. 2006. *Computational intelligence in time series forecasting: theory and engineering applications*. Springer Science & Business Media.

38. Predictive Analytics for Movie Openings 2013. VAST Mini Challenge 1: Visualize the box office. (2013). http://vacommunity.org/VAST+Challenge+2013, accessed December 2015.

39. Glenn Proctor and Chris Winter. 1998. Information flocking: Data visualisation in virtual worlds using emergent behaviours. In *Virtual Worlds*. Springer, 168–176.

40. Tobias Schreck, Tatiana Tekušová, Jörn Kohlhammer, and Dieter Fellner. 2007. Trajectory-based visual analysis of large financial time series data. *ACM SIGKDD Explorations Newsletter* 9, 2 (2007), 30–37.

41. Shobhit Seth. 2014. The Best Technical Analysis Trading Software. (2014). http://www.investopedia.com/articles/active-trading/121014/best-technical-analysis-trading-software.asp?header_alt=c, accessed December 2015.

42. Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*. 336–343.

43. Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents: A debate. *Interactions* 4, 6 (1997), 42–61.

44. K. Srinivas, B. Kavihta Rani, and A. Govrdhan. 2010. Applications of data mining techniques in healthcare and prediction of heart attacks. *International Journal on Computer Science and Engineering* 2, 02 (2010), 250–255.

45. Yoshiki Tanaka, Kazuhisa Iwamoto, and Kuniaki Uehara. 2005. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning* 58, 2 (2005), 269–300.

46. James J. Thomas and Kristin A. Cook. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society.

47. Jarke J. Van Wijk and Edward R. Van Selow. 1999. Cluster and calendar based visualization of time series data. In *Proceedings of IEEE Symposium on Information Visualization*. 4–9.

48. K. R. Venugopal, K. G. Srinivasa, and L. M. Patnaik. 2009. Fuzzy Based Neuro - Genetic Algorithm for Stock Market Prediction. In *Soft Computing for Data Mining Applications*. 139–166.

49. Marc Weber, Marc Alexa, and Wolfgang Müller. 2001. Visualizing Time-Series on Spirals. In *Proceedings of IEEE Symposium on Information Visualization*. 7–14.

50. Li Wei, Nitin Kumar, Venkata Lolla, Eamonn J. Keogh, Stefano Lonardi, and Chotirat Ratanamahatana. 2005. Assumption-Free Anomaly Detection in Time Series. In *Proceedings of the International Conference on Scientific and Statistical Database Management*. 237–242.

51. Michaël Zasso. 2014. ML-SOM Node.js Library. (2014). https://www.npmjs.com/package/ml-som, accessed December 2015.

52. Jian Zhao, Fanny Chevalier, and Ravin Balakrishnan. 2011a. KronoMiner: using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 1737–1746.

53. Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, and Ravin Balakrishnan. 2011b. Exploratory analysis of time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2422–2431.