

TIME SERIES PREDICTIVE MODEL FOR ESTIMATING METRIC FOR A GIVEN ENTITY

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] This application is an international patent application which claims the benefit of the filing date of U.S. Patent Application No. 63/145,404 filed February 3, 2021, which is herein incorporated by reference in its entirety for all purposes.

BACKGROUND

[0002] A multivariate time series can be used for estimating interaction metrics associated with entities in an interaction database. Estimating interaction metrics associated with an entities' transnational behavior within a payment processing network is essential for system monitoring. A multivariate time series, aggregated from an entities' past transaction history, can provide insight for estimating the interaction metric. The general multivariate time series prediction has been applied across several domains, including manufacturing, medical, and entomology. However, there are several domain related challenges associated with payment data, such as concept drift and multi-modality, in addition to the real-time requirements of handling payment data at scale.

[0003] Embodiments of the disclosure address this problem and other problems individually and collectively.

SUMMARY

[0004] One embodiment of the present invention includes a method performed by a computer. The method comprises receiving interaction data between electronic devices of a plurality of entities. The interaction data is used to form an entity interaction vector containing a number of interactions between the electronic devices of a chosen entity of the plurality of entities and the plurality of entities. The interaction data is additionally used to form an entity time series containing a plurality of metrics per unit time of the interactions of the electronic devices of the chosen entity. The computer can use an interaction encoder to generate an interaction hidden representation of the entity interaction vector using embeddings of the plurality of entities. The embeddings can represent interactions between

the electronic devices of the plurality of entities. The computer can use a temporal encoder to generate a temporal hidden representation of the entity time series. The temporal hidden representation can capture time-varying patterns of the interactions between the electronic devices of the plurality of entities. The interaction hidden representation and the temporal hidden representation can both be used to generate a predicted scale comprising a magnitude of a target interaction metric. The interaction hidden representation and the temporal hidden representation can both be used to additionally generate a shape estimation of the target interaction metric. The computer can then generate an estimated interaction metric of a time period using the predicted scale and the shape estimation.

5

10 **[0005]** Another embodiment of the present invention includes a computer. The computer comprises a processor and a non-transitory computer readable medium comprising instructions executable by the processor to perform operations including: receiving interaction data between electronic devices of a plurality of entities, wherein the interaction data is used to form (1) an entity interaction vector containing a number of interactions

15 between the electronic devices of a chosen entity of the plurality of entities and the plurality of entities and (2) an entity time series containing a plurality of metrics per unit time of the interactions of the electronic devices of the chosen entity; generating an interaction hidden representation of the entity interaction vector, wherein the interaction hidden representation captures information about interactions between the electronic devices of the plurality of

20 entities; generating a temporal hidden representation of the entity time series, wherein the temporal hidden representation captures inherent patterns of the interactions between the electronic devices of the plurality of entities; generating a predicted scale comprising a magnitude and an offset of a shape estimation of the interaction data of the interaction data; generating the shape estimation of the interaction data; and generating an estimated

25 interaction metric for a time period using the predicted scale of the interaction data and the shape estimation of the interaction data.

[0006] A better understanding of the nature and advantages of embodiments of the invention may be gained with reference to the following detailed description and accompanying drawings.

30

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] FIG. 1 shows an exemplary interaction time series for a plurality of entities.

[0008] FIG. 2 shows a two-dimensional projection of interaction vectors for a plurality of entities.

[0009] FIG. 3 shows a block diagram of an interaction metric estimation system according to embodiments of the present invention.

5 [0010] FIG. 4 shows a block diagram of an interaction metric estimation model according to embodiments of the present invention.

[0011] FIG. 5 shows a block diagram of an interaction encoder according to embodiments of the present invention.

10 [0012] FIG. 6 shows a block diagram of a temporal encoder according to embodiments of the present invention.

[0013] FIG. 7 shows a block diagram of a scale decoder according to embodiments of the present invention.

[0014] FIG. 8 shows a block diagram of a shape decoder according to embodiments of the present invention.

15 [0015] FIG. 9 shows a block diagram of an intermediate view of a shape decoder according to embodiments of the present invention.

[0016] FIG. 10 shows a block diagram of an amalgamate layer according to embodiments of the present invention.

20 [0017] FIG. 11 shows a training algorithm according to embodiments of the present invention.

[0018] FIG. 12 shows a modified FLUSS algorithm according to embodiments of the present invention.

[0019] FIG. 13 shows time series data, a corrected arc count, and an associated sampling probability according to embodiments of the present invention.

25 [0020] FIG. 14 shows a first baseline model according to embodiments of the present invention.

[0021] FIG. 15 shows a second baseline model including an entity interaction vector as input according to embodiments of the present invention.

[0022] FIG. 16 shows a table demonstrating the improvement of using an entity time series.

[0023] FIG. 17 shows a table demonstrating the improvement of using a shape and scale decoder.

5 [0024] FIG. 18 shows a table demonstrating the improvement of the interaction metric estimation model over several machine learning models.

[0025] FIG. 19 shows a table demonstrating the improvement of using online training in addition to offline training.

[0026] FIG. 20 shows a table demonstrating the performance benefits of improving the online training scheme.

10 [0027] FIG. 21 shows a table demonstrating an RMSE performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling techniques.

[0028] FIG. 22 shows a table demonstrating an NRMSE performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling
15 techniques.

[0029] FIG. 23 shows a table demonstrating an R^2 performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling techniques.

[0030] FIG. 24 shows a method for generating an estimated interaction metric for a time
20 period according to embodiments of the present invention.

[0031] FIG. 25 shows a block diagram of an exemplary computer system according to embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0032] Interactions can be performed by a variety of users of a network to send a receive
25 communications, and may occur across different geographical areas, such as countries. For example, users holding a credit card of a payment processing network may conduct transactions to shop for groceries, fine dining, or other to perform other purchases. Another example can include a user connecting via a first electronic device (e.g., a computer, a mobile device, a laptop, etc.) to a second electronic device to transmit data between the two
30 computers (e.g., the user can use the first computer to log in to an account maintained by the

second computer). The above examples are examples of access requests, where a user requests access to a resource from another computer.

[0033] To improve service and user experience, processing networks can study entity interaction behavior within the network. For each entity within the network, their interaction behavior can be modeled as a multivariate time series with multiple time varying features. Additionally, the entity specific multivariate time series can be further aggregated based on different criteria (e.g., where the credit card or account of a user of the payment network was issued) to provide additional macro-level insights. In some examples, the true interaction metrics may not be immediately available in the system due to limitations of the processing network, such as resource limitations in calculating the interaction metrics, delays in manual reports, or communication failures. In some examples, the true interaction metric may not be available to the processing network for a significant amount of time and as such it is imperative for the processing network to estimate the interaction metric for monitoring of the processing network. Significant challenges arise, including concept drift, the use of multiple modalities, and large interaction data sets, when building a model to estimate interaction metrics.

[0034] Concept drift occurs as the interaction behavior of entities performing interactions within a processing network is constantly evolving. A static prediction model can be rendered useless after days or months if the model does not actively evolve with the entities' behavior. Moreover, at a higher aggregation level, additional factors such as economic, geopolitical, and others (e.g., global pandemic) can have an impact on patterns within the multivariate time series of entities and the interaction metric, subsequently causing concept drift. Concept drift poses significant challenges in real-world applications, as opposed to experimental data which is static and well-cleaned. Online learning techniques are used to resolve the challenge of concept drift. Supervised online learning is a learning scheme wherein the training data is made available to the model incrementally over time.

[0035] A multi-modality approach that models the relationship, or the interactions among the entities of a network is preferred. As shown in FIG. 1, an entities' interaction time series may exhibit similar trends as another entity. If additional modalities further distinguish entities with similar time series, the model can use such modalities to better estimate the target entity-specific interaction metric. However, many conventional approaches lack the ability and means to utilize additional modalities.

- [0036] For many processing networks, due to the significant amount of interactions performed within the network (e.g., a payment processing network may conduct several millions of transactions a day), running hourly estimations for the interaction metric is not feasible. To overcome this limitation, a more realistic approach would be to predict multiple time-steps at once. This approach is known as a multi-horizon time series prediction, and requires a unique model design to handle such as situation. Applying a one-step prediction model in a rolling fashion can lead to inferior results, as the predictions for a later time step is made based on an estimated input. That is, errors from the earlier time step propagate through to later time steps.
- [0037] In some embodiments, a learning model capable of estimating multi-horizon interaction metrics using multi-modality of time series is used to handle the above two limitations of multi-modality and resource limitations. The learning model can comprise five unique components including an interaction encoder, a temporal encoder, a scale decoder, a shape decoder, and an amalgamate layer. The interaction encoder can be used to process the interaction modality (e.g., how entities interact with each other in the interaction data). The temporal encoder can dissect the temporal data and learns the inherent patterns of the interaction data. The scale decoder and the shape decoder can provide two distinct, yet related perspectives regarding the estimated multi-horizon interaction metric. The amalgamate layer can combine the outputs of the scale and shape decoder to synthesize the output.

I. ENTITY INTERACTIONS

- [0038] An interaction data set comprises data of interactions performed by an entity. One example of an interaction is an access request. An access request can be a first entity requesting access to a resource, such as a physical good or access to a computer account, from a second entity. The interaction data can contain the time that the interaction was performed. A time series of the interaction data set can be formed using the time of interactions. For example, an entity may be country and the interaction data set can be a transaction data set comprising transaction data of transactions performed by credit card users in the country. The transaction data can indicate what time the transaction was performed. A transaction time series may be formed using the time, and may indicate the number of transactions performed during each time step (e.g., transactions per minute, transactions per hour, etc.). Other examples of time series that can be formed using interaction data include

an interaction time series using a number of declined transactions during each time step, an amount of log in attempts to a computer account during each time step, an amount of transactions performed by a unique credit card held by users of the payment processing network during each time step or the like.

- 5 [0039] FIG. 1 shows an exemplary interaction time series for a plurality of entities. A first entity may be a first country, a second entity may be a second country, a third entity may be a third country, and a fourth entity may be a fourth country. The interaction time series for a country may include the number interactions performed in the country at a certain time. For example, the first entity time series 100 may be the amount of interactions performed in the
- 10 first country. Similarly, the second entity time series 102, the third entity time series 104, and the fourth entity time series 106 can be the interaction time series for second country, a third country, and a fourth country respectively. Similar countries (e.g., based on location) may exhibit behaviors. For example, the first entity time series 100 and the second entity time series 102 more closely resemble each other time series, and so they may be geographically
- 15 close (e.g., the first entity maybe Australia and the second entity may be New Zealand).

II. INTERACTION MODELING

- [0040] An entity can be represented from interaction data with several views. One view includes an entity interaction time series. An entity interaction time series for an entity E is a multivariate time series, denoted as $T_E \in \mathbb{R}^{\tau_e \times d}$, where τ_e is the length of the time series and
- 20 d is the number of features. $T_E[i:j]$ can denote the subsequence starting at the i -th timestamp, and ending at the j -th timestamp.

- [0041] In an example system, each entity may be a country and the interaction data statistics are computed hourly. A prototype model is trained using transaction data of several countries from January 1st, 2017 through December 31st, 2017. The length of the time series
- 25 τ_e in the transaction data is equal to $24 \times 365 = 8,760$. The number of features d is the transaction data is equal to 14, which is the number of statistics of the transaction data that is extracted during the hourly window.

- [0042] Another view of an entity includes an entity interaction vector. Given an entity E within an entity set $\mathbb{E}$ containing k total entities, the entity interaction vector $I_E \in \mathbb{R}^k$ is
- 30 defined as the amount of interaction between the entity E and each of the other k total

entities. The entity interaction vector I_E can vary over time, and $I_E[i]$ can denote the entity interaction vector at the i -th timestamp.

[0043] In the global transaction data used for training there are 233 total counties. The entity interaction vector for a country i is a vector of length 233. Each entity interaction
 5 vector records the number of transactions made by cards (e.g., credit cards) issued in the country i , at merchants with different countries within the past thirty days. The entity interaction vector is computed daily, meaning there is a total of 365 entity interaction vectors for each country in the transaction data. The entity time series T_E captures the entities' E behavior at a finer time resolution (e.g., hourly rather than daily), whereas the entity
 10 interaction vector I_E provides a view on the relationship between different countries in a coarser time resolution (e.g., a thirty day window). Each country's interaction vector is projected to two-dimensions for the first month of 2017 and is shown in FIG. 2.

[0044] FIG. 2 shows a two-dimensional projection of interaction vectors for a plurality of entities. Geographically close countries (e.g., Mexico, Canada, and the United States of
 15 America) are generally close in the plot 200. The plot 200 of all country interaction vectors captures both geographic and political relationships amongst the countries of the global transaction data used to train the prototype model. Concept drift can be seen by the movement of the countries in the plot 200. For example, a merchant in Canada that processes transactions with a payment processing network may choose to no longer purchase goods
 20 from a supplier in France, and instead purchase goods from a different supplier in Germany. Overtime, changes in the transaction behavior of merchants cause concept drift.

[0045] An entity interaction metric is a time series, denoted as $M_E \in \mathbb{R}^{\tau_m}$, where τ_m is the length of the time series. The entity interaction metric M_E can vary with time, and $M_E[i:j]$ denote the subsequence starting at the i -th timestamp and ending at the j -th timestamp. For
 25 the systems and models presented in the Application, the entity interaction metric series stores each of the entities' target interaction metric per time period. As both M_E and T_E have the same sampling rate, both time series have the same length (e.g., the length τ_e is equal to 8,760 for the global transaction data, and is equal to the length τ_m).

III. INTERACTION METRIC ESTIMATION SYSTEM

30 [0046] Given an entity E , the entity interaction vector $I_E[i]$ and the entity time series $T_E[i - t_p:j]$, the goal of the interaction metric estimation is to learn a model F that can be

used to predict the interaction metric of the entity between the time $i - t_a$ and $i + t_b$. t_a is the number of backward time steps and t_b is the number of forward time steps desired to estimate the interaction metric. The model F can be formulated as follows:

$$\hat{M}_E[i - t_p : i + t_b] \leftarrow F(T_E[i - t_p : i], I_E[i]) \quad (1)$$

5 [0047] Where $\hat{M}_E[i - t_p : i + t_b]$ is the estimated interaction metric from $i - t_a$ and $i + t_b$.

[0048] A multi-horizon time series is used in the estimation model F to create buffer time between each consecutive prediction to ensure that there is no downtime in the production environment. As a running example, the time period that the interaction metric is predicted is equal to 24 hours (e.g., $t_b = 24$). For some interaction metrics, estimating the past metric is also required if there is a delay before the model F can access the true interaction metric. By
10 observing an entities' past interaction behavior, the model F can generate a more accurate estimation for analysis before the true interaction metric is available to the model F . The estimation model F estimates the interaction metric for the past 24 hours (e.g., $t_a = 24$). For the input entity time series, only t_p time steps are used instead of all of the available time
15 series for efficiency. In the running example, a total of 168 hours (e.g., $t_a = 168$), or seven days, is used as the time step t_p . The past target metric is not a part of the input to the estimation model F because the delay before the model F can observe the true interaction metric is longer than seven days.

[0049] Given a training data set consisting of both entity time series and entity interaction
20 vectors from time step 1 through to time step τ_e for each entity E in the entity set $\mathbb{E}$, the model F is learned by minimizing the following loss function:

$$\sum_{E \in \mathbb{E}, i \in [1, \tau_e]} \text{loss}(\hat{M}_E[i - t_a : i + t_b], M_E[i - t_p : i + t_b]) \quad (2)$$

Where $\text{loss}()$ is can be any regression loss function, such as mean squared error.

A. System Overview

25 [0050] The interaction metric estimation system has two training phases: a first offline training phase and an online training phase. Before the interaction metric estimation system is deployed, an offline training module may pull data from an interaction database to train an initial interaction metric estimation model.

[0051] FIG. 3 shows a block diagram of an interaction metric estimation system according to embodiments of the present invention. The interaction metric estimation system comprises an interaction computer 300, an interaction database 302, an offline training module 304, an online training module 306, an estimation module 308, and a GUI display 310. The interaction computer 300 may be operated by a processing network such as a payment processing network. The interaction computer 300 may be configured to receive information about interactions between the electronic devices of the plurality of entities. The interaction computer 300 may communicate with electronic devices of a plurality of entities to perform interactions.

10 [0052] The interaction computer 300 may receive an access request for a destination entity of the plurality of entities. The access request can comprise interaction data for an interaction between a sender entity and a receiver entity. The interaction data of the access request can be used to update the entity interaction vector and the entity time series associated with the destination entity of the plurality of entities. For example, the interaction computer 300 may
15 communicate with an access device (e.g., a POS terminal) operated by a merchant located in a first country (e.g., a receiver entity) and an issuer computer, associated with a credit card of a user, located in a second country (e.g., a sender entity) to perform a transaction between the merchant and the user to receive transaction data related to the transaction. The access device and the issuer computer (or the credit card) can be examples of electronic devices of a
20 plurality of entities.

[0053] At step 312, the interaction computer 300 can store interaction data in the interaction database 302. The interaction data may comprise a number of features on the interactions (e.g., the type of interaction that was made, the location the interaction was conducted at, the time the interaction was conducted). The interaction data may be
25 aggregated based on the time the interaction was conducted to form entity interaction vectors and interaction time series relating to the entity that performed the interaction. For example, the interaction computer 300 can use the interaction data received to modify an existing time series and interaction vector associated with the entity that performed the interaction. In another example, the interaction computer 300 can use all of the stored interaction data
30 associated with an entity to generate an entity time series and an entity interaction vector.

[0054] At step 314, the offline training module 304 may pull interaction data from the interaction database 302. The offline training module 304 may use the interaction data to

train an initial interaction metric estimation model. The entity interaction vectors and entity interaction time series may be by the offline training module 304 to train the initial interaction metric estimation model using an offline training algorithm.

5 **[0055]** The interaction computer 300 may continue to process interactions and update the interaction data held by the interaction database 302 to include the new interactions. The new interactions may be used to update entity interaction vectors and time series as they are received.

10 **[0056]** At step 316, after the initial interaction metric estimation model is trained, the online training module 306 may retrieve the initial interaction metric estimation model from the offline training module 304.

[0057] At step 318, the online training module 306 can retrieve the updated interaction data from the interaction database 302. The online training module 306 can update the initial interaction metric estimation model using the updated interaction data using an online training algorithm.

15 **[0058]** Additionally, at step 318, the estimation module 308 can retrieve the updated interaction data from the interaction database 302. At step 320, after the online training module 306 updates the initial interaction metric estimation model, the estimation module 308 may retrieve the updated interaction metric estimation model from the online training module 306. The estimation module 308 may produce the estimated interaction metric for a
20 next time period and a previous time period for an entity. For example, the estimation module 308 can produce the estimated interaction metric for an entity for both the previous 24 hours and the next 24 hours.

[0059] At step 322, after producing the estimated interaction metric for an entity can be stored by the estimation module 308 in the interaction database 302.

25 **[0060]** At step 324, the estimated interaction metric in the interaction database 302 can be displayed on the GUI display 310. For example, the GUI display 310 can display a graphical representation of the most recent estimated interaction metric. A user of the interaction metric estimation system can monitor the system by visually viewing the graphical representation, or by comparing the estimated interaction metric with the true interaction
30 metric when it is available (e.g., calculating the difference between the estimated interaction metric and the true interaction metric). The estimated interaction metric can be fed to

external modules of the interaction computer 300. For example, the estimated interaction metric can be fed to a fraud detection module, so that the fraud detection module can be trained using the estimated interaction metric to better detect fraudulent access requests (e.g., to detect fraudulent transactions or log in attempts). Another example can include an evaluation module that calculates the difference between the estimated interaction metric and the true interaction metric. The evaluation module can identify time periods where the estimated interaction metric and the true interaction metric differ significantly, so that the user of the interaction metric estimation system may more closely analyze the interaction data during the identified time periods. Steps 316 through 324 may occur as part of a daily routine, as shown by the dotted box 350.

B. Example Model

[0061] The interaction metric estimation model in the above interaction metric estimation system is used to generate the estimated interaction metric. The interaction metric estimation model comprises learnable parameters that can be updated based on new interaction data received.

[0062] FIG. 4 shows a block diagram of an interaction metric estimation model 400 according to embodiments of the present invention. The interaction metric estimation model 400 consists of five components, including an interaction encoder 404, a temporal encoder 408, a scale decoder and shape decoder 412, and an amalgamate layer 412. The interaction metric estimation model 400 may take an entity interaction vector, I_E 402 and an entity time series, T_E 406 as input.

[0063] The interaction encoder 404 can process the entity interaction vector, I_E 402. In the example shown in FIG. 4, the entity interaction vector, I_E 402 is of length k equal to nine. The entity represented by the vector would thus interact with nine total entities (e.g., nine different countries). The output of the interaction encoder 404 can be an interaction hidden representation $h_I \in \mathbb{R}^{n_k}$, where n_k is the embedding size for each entity. The embeddings can represent (e.g., by capturing information about) the interactions between electronic devices of different entities in an interaction database, and when combined (e.g., using vector-matrix multiplication) with the interaction vector 402, the resulting interaction hidden representation h_I can capture the information about the interactions between different entities in an interaction database. Multiple interaction types can be integrated by increasing the amount of interaction encoders 404 in the interaction metric estimation model 400.

[0064] The temporal encoder 408 can extract the temporal hidden representation $\mathbf{h}_T \in \mathbb{R}^{n_k}$ from the input entity time series, $\mathbf{T}_E \in \mathbb{R}^{t_p \times d}$ 406, where n_k is size of the output hidden representation vector, t_p is the length of the input time series, and d is the dimensionality of the input time series.

- 5 [0065] The scale/shape decoder 412 combine the temporal hidden representation $\mathbf{h}_T$ with the interaction hidden representation $\mathbf{h}_I$ to generate a scale (e.g., a magnitude σ and an offset μ) and a shape estimation for the interaction metric.

[0066] The amalgamate layer 412 generates the estimated interaction metric, $\mathbf{M}_E$ 414 by processing the shape estimation with the scale obtained from the scale/shape decoder 412.

10 IV. MODEL ARCHITECTURE

- [0067] The two encoders (e.g., the interaction encoder 404 and the temporal encoder 408) extract hidden representations (e.g., the interaction hidden representation $\mathbf{h}_I$ and the temporal hidden representation $\mathbf{h}_T$) from the inputs (e.g., the entity interaction vector, $\mathbf{I}_E$ 402 and the entity time series, $\mathbf{T}_E$ 406) independently. Each encoder is only responsible for one aspect of
- 15 the input entity. Using the two extracted hidden representations, each decoder then independently provides information regarding different aspects of the estimated interaction metric. The scale decoder provides the scale information (e.g., the magnitude σ and the offset μ), while the shape decoder provides the shape information. The amalgamate layer 412 combines the shape and scale information to generate the estimated interaction metric,
- 20 $\mathbf{M}_E$ 414. Further details of each component of the interaction metric estimation model 400 are described below.

A. Interaction Encoder

- [0068] FIG. 5 shows a block diagram of an interaction encoder 500 according to embodiments of the present invention. The input to the interaction encoder 500 can be an
- 25 entity interaction vector, $\mathbf{I}_E$ 502. The entity interaction vector, $\mathbf{I}_E$ 502 of an entity stores information about the relationship between the chosen entity E and other entities. The entity interaction vector, $\mathbf{I}_E$ 502 can be normalized through a sum-to-one normalize module 504. For example, as shown in FIG. 5, the entity interaction vector, $\mathbf{I}_E$ 502 [0, 5, 0, 1, 0, 0, 3, 0, 1] can be normalized to obtain the normalized interaction vector 506 [0, 0.5, 0, 1, 0, 0, 0.3, 0, 1] such that the magnitude of the normalized interaction vector 506 is equal to one. The sum-to-
- 30

one normalize module 504 is used for modeling interactions between different entities in an interaction database. For example, in the country-level transaction metric estimation problem used as a running example, the L1 vector norm of the entity interaction vector, $\mathbf{I}_E$ 502 is proportional to the population of the country. The sum-to-one normalization ensures that the hidden representation focuses on capturing the information regarding the interaction between different countries, rather than the population difference of countries.

[0069] The normalized interaction vector 506 is combined with an entity embedding matrix, $\mathbf{C}$ 508 to generate the interaction hidden representation $\mathbf{h}_I$ 512 (a stubbed version is shown in FIG. 5). A vector-matrix multiplication module 510 may compute the interaction hidden representation, $\mathbf{h}_I = \bar{\mathbf{I}}_E \mathbf{C}$ 512. The entity embedding matrix, $\mathbf{C}$ 508 contains embeddings corresponding to each country. The interaction hidden representation, $\mathbf{h}_I$ 512 is thus a weighted sum of the embeddings of the entity embedding matrix, $\mathbf{C}$ 508. The embeddings of the entity embedding matrix, $\mathbf{C}$ 508 is a learnable parameter of the interaction encoder 500. The entity embedding matrix, $\mathbf{C}$ 508 may be initialized randomly, or using existing embeddings. The size of the interaction hidden representation, $\mathbf{h}_I$ 512 is dependent on each country's embedding vector size. In the running example, the total number of countries is equal to 233, and the embedding vector is set to 64.

B. Temporal Encoder

[0070] FIG. 6 shows a block diagram of a temporal encoder 600 according to embodiments of the present invention. The input to the temporal encoder 600 can be an entity time series, $\mathbf{T}_E \in \mathbb{R}^{t_p \times d}$ 602, where t_p is the length of the input time series, and d is the dimensionality of the input time series. The output of the temporal encoder 600 can be a temporal hidden representation vector, $\mathbf{h}_T \in \mathbb{R}^{n_k}$ 640, where n_k is the vector size of the output. The temporal encoder 600 analyzes the temporal patterns by extracting features and reducing the dimensionality of the input entity time series, $\mathbf{T}_E$ 602. The temporal hidden representation vector, $\mathbf{h}_T$ 640 captures the temporal patterns of the entity time series, $\mathbf{T}_E$. The vector size n_k depends on the hyper-parameter setting of the last convolutional layer (e.g., the size of the kernel of the convolutional layer). In the running example, t_p is equal to 168 (e.g., seven days of hourly data), and d is equal to 14.

[0071] In this example, the body of the temporal encoder 600 comprises a sequence of identical residual blocks. As an example, all of the convolution layers in the temporal

encoder 600 except the first convolution layer, Conv 604, have a kernel size of three. Similarly, all convolution layers except the first have 64 kernels. The main passage (e.g., the path of Conv 606 to ReLU 608 to Conv 610 to ReLU 612, or the path of Conv 616 to ReLU 618 to Conv 620 to ReLU 622, etc.) is processed with Conv-ReLU-Conv-ReLU layers before passing through a ReLU layer (e.g., ReLU 614, or ReLU 624). The first residual block (e.g., layers beginning from Conv 606 through ReLU 614) has a convolution layer, Conv 604, that is used to solve the mismatch between the output of the main passage and the residual passage. For example, the input of the residual passage for the first block has 14 channels (e.g., the dimensionality d of the entity time series, T_E 602). The convolutional layers in the residual passage convert the input to 64 channels, which matches the output of the main passage (e.g., the number of kernels for the last convolution layer, Conv 610 of the main passage). The kernel size for the first convolution layer, Conv 604, is equal to one. The output temporal hidden representation vector, h_T 640 is generated by summarizing the output of the residual blocks across time with the global average pooling given by the AvgPool layer 638. The length of the temporal hidden representation vector, h_T 640 matches the number of kernels of the last convolution layer, Conv 632, which is equal to 64.

C. Scale Decoder

[0072] FIG. 7 shows a block diagram of a scale decoder 700 according to embodiments of the present invention. The input to the scale decoder 700 can be a temporal hidden representation vector, h_T 702 and a interaction hidden representation, h_I 712. The Linear 704 to ReLU 706 to Linear 708 to ReLU 710 layers on the left further process the temporal hidden representation vector, h_T 702 to focus on the information relevant for predicting the scale of the interaction metric. The Linear 714 to ReLU 716 to Linear 718 layers on the right produce a matrix $W \in \mathbb{R}^{n_k \times 2}$ that is used for mapping the output of the left layers to generate a magnitude σ and an offset μ . Entities with similar interaction representations use similar matrices W to estimate the scale of the target interaction metric.

D. Shape Decoder

[0073] FIG. 8 shows a block diagram of a shape decoder 800 according to embodiments of the present invention. The input to the shape decoder 800 can be an interaction hidden representation vector, h_I 802 and a temporal hidden representation, h_T 812. The structure of the shape decoder 800 is similar to the structure of the scale decoder 700. The output of the

Linear 804 to ReLU 806 to Linear 808 layers on the left operate on the interaction hidden representation vector, $\mathbf{h}_T$ 802 to generate a shape bank 810. The shape bank 810 stores basis shapes for estimating the shape of the target interaction metric. The right leg consists of a Linear 814 to ReLU 816 to Linear 818 to Softmax 820 layer. The last layer for processing the temporal hidden representation, $\mathbf{h}_T$ 812 is a Softmax 820 layer, instead of the ReLU 710 layer of the scale decoder 700. The output of the Softmax 820 layer dictates which, and how the shape bank's basis shapes are combined to form the shape prediction 822. For example, the output provides the weights that will be used to combine basis shapes in the shape bank 810. Entities with similar interaction representations should use similar basis shapes for estimating the shape of the target transaction metric. FIG. 9 provides further details on the shape bank 810.

[0074] FIG. 9 shows a block diagram of an intermediate view of a shape decoder according to embodiments of the present invention. The intermediate output 900 may be the output of the ReLU 816 layer of the shape decoder 800. The remaining layers (e.g., the Linear 818 and Softmax 820 layers) form a Softmax regression model. The output of the Softmax regression model 906 is always positive and always sums-to-one, it forces the model to pick only relevant basis shapes from the shape bank 908 for the shape estimation 910. As shown in FIG. 9, the shape estimation 910 is generated by combining the basis shapes in the shape bank 910 (e.g., the first shape has a weight of 0.5, the second shape has a weight of 0.4, and the third shape has a weight of 0.1, etc.).

E. Amalgamate Layer

[0075] FIG. 10 shows a block diagram of an amalgamate layer 1000 according to embodiments of the present invention. The amalgamate layer 1000 combines a shape estimation 1002 from a shape decoder with a predicted scale 1004 from a scale decoder. The predicted scale 1004 can comprise a magnitude σ and an offset μ . The shape estimation 1002 can be multiplied by the magnitude σ , then the offset μ can be added to the shape estimation 1002. The following loss function (adapted from equation (3)) can be minimized to improve the estimated interaction metric, $\hat{\mathbf{M}}_E$ 1006:

$$\sum_{E \in \mathbb{E}} \sum_{t \in [1, \tau_e]} \text{MSE}(\hat{\mathbf{M}}_E, \mathbf{M}_E) + \gamma \text{NMSE}(\hat{\mathbf{M}}_E^{\text{shape}}, \mathbf{M}_E) \quad (3)$$

Where $\text{MSE}()$ is a function that computes the mean squared error, $\text{NMSE}()$ is a function that computes the normalized mean squared error, γ is a hyper-parameter to ensure the output of

MSE() and NMSE() are of a similar scale, $\hat{\mathbf{M}}_E$ is the estimated interaction metric, $\hat{\mathbf{M}}_E$ 1006, and $\hat{\mathbf{M}}^{shape}_E$ is the output of the shape decoder. The normalized mean square error is computed by z-normalizing the ground truth, then calculating the mean squared error between the z-normalized ground truth and the estimated interaction metric, $\hat{\mathbf{M}}_E$ 1006.

5 V. ONLINE LEARNING SCHEME

[0076] As described in section III.a., the interaction metric estimation model can be trained to improve the estimated interaction metrics. Training of the interaction metric estimation model can comprise an offline training phase and an online training phase.

A. Training Algorithm

10 [0077] FIG. 11 shows a training algorithm 1100 according to embodiments of the present invention. The training algorithm 1100 takes as input a set of entities $\mathbb{E}$. The line 2, the interaction metric estimation model is initialized. Next, in line 3, the interaction metric estimation model is trained offline using available interaction data (e.g., as described in step 314 of FIG. 3). In line 4, an online training phase can be initiated. In lines 5 and 6, the
 15 function can check if it is time to update the model. For example, if the interaction metric estimation model is updated daily, the time to update variable can be a variable that counts 24 hours. In line 7, after determining the interaction metric estimation model is to be updated, the latest interaction data (e.g., determined by the current value of the time to update variable) can be pulled from an interaction database. In lines 8 through 10, the model can be updated
 20 for n_{iter} total iterations. In line 9, a batch of entities and their associated interaction vectors, time series, and interaction metric can be sampled from the set of entities. The sampling step has a significant impact on the performance of the interaction metric estimation model. When concept drift is encountered, irrelevant data is encountered more often. In line 10, the interaction metric estimation model is updated using the mini-batch. In line 11, the estimated
 25 interaction metric for each entity in the set of entities is generated and the estimated interaction metrics are returned in line 12.

B. Modified FLUSS Algorithm

[0078] Improvements to the sampling method for the mini-batch can be made to counter concept drift. Time series segmentation can be used to identify, in a data driven fashion, a
 30 window of data to be used in the mini-batch, and a time decay function. Several sampling

techniques can be used for sampling the mini-batch when updating the model during online training. Two categories of sampling techniques, including temporal-based and non-temporal-based sampling methods are identified.

[0079] Temporal-based sampling techniques can include fixed window sampling, time decay sampling, and time series segmentation sampling. Fixed window sampling samples the training examples within the latest x days uniformly, where x is a hyper-parameter (e.g., it ignores older data). Time decay sampling samples the training examples with a decaying probability as the data ages. In later examples, time decay sampling uses linear decay. Segmentation sampling uses a window and time decay function in a data driven fashion, based on a matrix profile. The matrix profile includes nearest neighbor relationship between subsequences of a time series sequence.

[0080] Non-temporal sampling techniques can include similarity, high error, low error, and training dynamic based sampling. In non-temporal sampling techniques, the temporal location of each candidate sample is not considered in the sampling process. Similarity biases towards examples that are more similar to the current time series (e.g., $T_E[i - t_p : i]$, where i is the current time). As it only looks at the current time series it helps when concept drift affects the time series. High error biases towards “hard” examples for the current interaction metric estimation model. Pushing the interaction metric estimation model towards hard examples is commonly seen in boosting-based ensemble methods. Low error biases towards examples that can be predicted well based on the current model. The targeted interaction metric can be noisy, and low error biases can be used to remove noisy samples as they tend to introduce large errors. Training dynamic based sampling uses confidence and variability to sample data.

[0081] FIG. 12 shows a modified FLUSS algorithm 1200 according to embodiments of the present invention. The modified FLUSS algorithm 1200 is built upon a matrix profile. The matrix profile can be used to efficiently explore the nearest neighbor relationship between subsequence of a time series sequence. The input to the modified FLUSS algorithm 1200 is an entity time series, T_E . In lines 2 through 5, each iteration of the loop processes each dimension of the entity time series, T_E independently. In line 3, the matrix profile index of the input time series i -th dimension is computed. The matrix profile index shows the nearest neighbor of each subsequence in the input entity time series, T_E , where the subsequence length used is t_p . Each subsequence is connected with its nearest neighbor using an arc

based on the information in the matrix profile index, as shown in FIG. 13. To further process the matrix profile index, the corrected arc count curve is computed in line 4. The corrected arc count curve records the number of arcs passing through each temporal location in the time series. The corrected arc count curve includes a correction to correct that it is more likely for an arc to pass through the center of the time series than an arc to pass through the ends of the time series. The correction is applied by comparing the actual count of arcs to the expected count. Once the correct arc count curve is computed, the corrected arc count curve is converted to a sampling probability. From lines 6 to 11, a for loop is used to enforce a non-decreasing constrain to ensure the subsequences belong to the newest regime (e.g., subsequences after the latest segmentation point) have higher sampling probabilities compared to subsequences from the old regimes. In line 12, the non-decreasing corrected arc count curve is converted to a probability and is returned in line 13. The probability can be used in line 9 of the training algorithm 1100 to more efficiently sample the mini-batch.

[0082] FIG. 13 shows time series data 1300, a corrected arc count 1302, and an associated sampling probability 1304 according to embodiments of the present invention. The first and second half of the time series 1300 are significantly different. An arc only connects similar subsequences, and as such there is almost no arc passing through the center of the time series that connects similar subsequences. For this reason, the corrected arc count 1302 sees a decline near the regime change. The corrected arc count 1302 is converted into a sampling probability 1304. A non-decreasing constrain (e.g., lines 6 through 11 of the modified FLUSS algorithm 1200) is applied to the corrected arc count 1302 to ensure the subsequences belonging to the newest regime have higher sampling probabilities. The non-decreasing constrain flattens the corrected arc count 1302 before the regime change to form the sampling probability 1304.

25 VI. EVALUATION OF MODEL

[0083] The interaction metric estimation model provides a number of advantages. The interaction metric estimation model allows a processing network to more efficiently and more accurately predict interaction metrics. The design of the example model shown in FIG. 18 result in a number of improvements over conventional methods.

A. Model Verification

- [0084] FIG. 14 shows a first baseline model 1400 according to embodiments of the present invention. The first baseline model 1400 can take an entity time series, T_E 1402 as input. The first baseline model 1400 does not include an interaction vector. Further, the first
- 5 baseline model 1400 is simpler than the interaction metric estimation model shown in FIG. 4, as it does not include either the shape or scale decoders.
- [0085] FIG. 15 shows a second baseline model 1500 including an entity interaction vector, I_E 1514 as input according to embodiments of the present invention. The second baseline model 1500 is simpler than the interaction metric estimation model shown in FIG. 4, as it
- 10 does not include either the shape or scale decoders. The second baseline model 1500 is trained using equation (2), by minimizing the MSE loss function. The passage that processes the entity interaction vector, I_E 1514 outputs a matrix $W \in \mathbb{R}^{n_k \times (t_a + t_b)}$. The matrix W is multiplied by the output of the passage that processes the entity time series, T_E 1602 to generate the estimated interaction metric, $\hat{M}_E$ 1624.
- 15 [0086] FIG. 16 shows a table 1600 demonstrating the improvement of using an entity time series. The temporal encoders of the first baseline model 1400 and the second baseline model 1500 can be based on a convolutional neural network (CNN), as shown in FIG. 6, or they can be based on two layers of a recurrent neural network (RNN) with gated recurrent units (GRUs). There is a consistent improvement seen by using the second baseline model 1500
- 20 over the first baseline model 1400. The performance measures used include root mean squared error (RMSE), normalized root mean squared error (NRMSE), and coefficient of determination (R^2). The improvements are consistent through the use of both temporal encoders based on CNNs and RNNs, with RNN-based temporal encoders performing slightly better than the CNN-based temporal encoder.
- 25 [0087] FIG. 17 shows a table 1700 demonstrating the improvement of using a shape and scale decoder. The second baseline model 1500 of FIG. 15 is compared to the interaction metric estimation model 400 of FIG. 4. Again, the improvement of using a shape and scale decoder is consistent over the use of RNN-based and CNN-based temporal encoders and over different performance measures. The best performance, according to table 1700, is when the
- 30 interaction metric estimation model uses a CNN-based temporal encoder (e.g., the interaction metric estimation model 400).

[0088] FIG. 18 shows a table 1800 demonstrating the improvement of the interaction metric estimation model over several machine learning models. The interaction metric estimation model 400 is compared to several off-the-shelf machine learning solutions in a production environment. The percent improvement is computed against the best alternative, based on each performance measurement. As both NRMSE and R^2 are computed on a normalized scale and measure how the estimated trend matches the ground truth, the interaction metric estimation model captures more details in the interaction metric than the alternative machine learning models.

A. Online Learning

[0089] FIG. 19 shows a table 1900 demonstrating the improvement of using online training in addition to offline training. The interaction metric estimation model 400 of FIG. 4 is used to verify the benefits of online training. The offline training row calculates performance metrics of the interaction metric estimation model when it is only trained once, using interaction data stored in an interaction database. The online training row calculates performance metrics of the interaction metric estimation model that is updated using the training algorithm 1100 of FIG. 11. Online training provides an improvement over only using offline training in all performance metrics used, and provides a significant improvement in the R^2 performance metric.

[0090] FIG. 20 shows a table 2000 demonstrating the performance benefits of improving the online training scheme. Refining the online training scheme used to train the interaction metric estimation model can provide further increases in performance metrics. A baseline online training scheme (e.g., uniform sampling of the time series) is compared to an improved online training scheme that implements segmentation and similarity when sampling the time series. Increases in all three performance metrics are seen.

[0091] Temporal-based sampling techniques (e.g., uniform sampling, 90 day fixed window sampling, 365 day fixed window sampling, time decay sampling, and time series segmentation sampling) and non-temporal based sampling techniques (e.g., uniform, similarity, high error, low error, high-confidence, low-confidence, high-variability, and low variability) can be used in the online learning scheme. Temporal-based sampling techniques and non-temporal based sampling techniques are combined.

[0092] FIG. 21 shows a table 2100 demonstrating an RMSE performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling techniques. Generally, to learn a better interaction metric estimation model, it needs to see interaction data more than 90 days old, as the 90 days fixed window performs worse than the baseline. For non-temporal based methods, similar, low-error, high-confidence, and low-variability outperform the baseline. The low-error, high-confidence, and low-variability push the interaction metric estimation model push the model to focus on easy or consistent examples in the training interaction data. The improvement is likely caused by the removal of noisy training examples. Combining low-error sampling with segment sampling results in the best RMSE performance metric.

[0093] FIG. 22 shows a table 2200 demonstrating an NRMSE performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling techniques. According to the table 2200, for temporal-based methods, segment gives superior performance compared to others. When combining segment with non-temporal-based methods, the conclusion is similar to table 2100. Better methods include similar, low-error, high-confidence, and low-variability. Overall, combining similar with segment results in the best NRMSE performance metric.

[0094] FIG. 23 shows a table 2300 demonstrating an R^2 performance metric of several combinations of temporal-based sampling techniques and non-temporal-based sampling techniques. R^2 is more similar to NRMSE compared to RMSE, and as such when examining only temporal-based sampling techniques or non-temporal-based sampling techniques results in the same conclusions as table 2200. The combination of similar with 90 day fixed window results in the best R^2 performance metric.

[0095] Considering table 2100, table 2200, and table 2300, both of the combination of similar sampling with segment sampling, and the combination of low-error bias with segment sampling have great average ranks in all performance metrics. The interaction metric estimation model 400 of FIG. 4 was presented using similar sampling with segment sampling.

VII. METHOD

[0096] FIG. 24 shows a method 2400 for generating an estimated interaction metric for a time period according to embodiments of the present invention. The estimated interaction

metric may provide an estimate for a true interaction metric over the time period. For example, the estimated interaction metric may estimate the true interaction metric for the next 24 hours. The method 2400 may be performed by a computer, such as the interaction computer 300 of FIG. 3. The method 2400 may be performed by the computer after each
5 time period (e.g., if the time period is 24 hours, the computer may perform the method daily so that the interaction metric is estimated daily). The computer can choose the time period that the estimated interaction metric is generated for by training an interaction metric estimation model comprising an interaction encoder, a temporal encoder, a scale decoder, a shape decoder, and an amalgamate layer. For example, the computer can use the algorithms
10 of FIGs. 11 and 12 using interaction data nearest to the chosen time period.

[0097] At step 2402, the computer can receive interaction data between electronic devices of a plurality of entities. The interaction data can be used to form (1) an entity interaction vector (e.g., as described in section II) and (2) an entity time series (e.g., as described in section I). The entity interaction vector provides a view of the relationship between the
15 plurality of entities. The entity interaction vector can contain a number of interactions between the electronic devices of a chosen entity of the plurality of entities and the plurality of entities. The entity time series can contain a plurality of metrics per unit time the interactions of the electronic devices of the chosen entity. Examples of metrics can include number of interactions (e.g., access requests) per unit time (e.g., a number of transactions
20 performed, a number of log in attempts made in to a computer account), a number of declined interactions per unit time (e.g., a number of declined transactions), or a number of unique accounts used to perform interactions per unit time (e.g., a number of unique credit cards made to perform a transaction, a number of unique accounts that were logged in to).

[0098] At step 2404, the computer can generate an interaction hidden representation of the
25 entity interaction vector. The interaction hidden representation captures information about interactions between the electronic devices of the plurality of entities. The computer can generate the interaction hidden representation of the entity interaction vector using an interaction encoder that uses the entity interaction vector and an entity embedding matrix to generate the interaction hidden representation. The entity embedding matrix can contain
30 embeddings corresponding to each of the plurality of entities. For example, the computer can use the interaction encoder 500 of FIG. 5 with the entity interaction vector as input to generate the interaction hidden representation of the entity interaction vector. As described in FIG. 5, the interaction encoder can comprise a normalization module that operates on the

entity interaction vector to generate a normalized entity interaction vector. Further, a vector-matrix multiplication module that operates on the entity embedding matrix and the normalized entity interaction vector can be used to generate the interaction hidden representation.

5 [0099] At step 2406, the computer can generate a temporal hidden representation of the entity interaction vector. The temporal hidden representation captures inherent patterns of the interactions between the electronic devices of the plurality of entities. The computer can generate the temporal hidden representation of the entity time series using a temporal encoder that uses the entity time series to generate the temporal hidden representation. For example,
10 the computer can use the temporal encoder 600 of FIG. 6 with the entity time series as input to generate the temporal hidden representation of the entity time series. As described in FIG. 6, the temporal encoder can comprise a plurality of residual blocks and an average pooling layer that operate on the inputted entity time series.

[0100] At step 2408, the computer can generate a predicted scale comprising a magnitude
15 and offset of a shape estimation of the interaction data of the interaction data. The predicted scale can comprise a magnitude and an offset of a shape estimation of the interaction data. The computer can generate the predicted scale of the interaction data using a scale decoder that uses the interaction hidden representation and the temporal hidden representation to generate the predicted scale. For example, the computer can use the scale decoder 700 of
20 FIG. 7 with the interaction hidden representation and the temporal hidden representation to generate the predicted scale comprising the magnitude and the offset of the shape estimation of the interaction data. As described in FIG. 7, the scale decoder can comprise a plurality of linear activations layers and rectifiers.

[0101] At step 2410, the computer can generate the shape estimation of the interaction data.
25 The computer can generate the shape estimation of the interaction data using a shape decoder that uses the interaction hidden representation and the temporal hidden representation to generate the shape estimation. For example, the computer can use the shape decoder 700 of FIG. 7 with the interaction hidden representation and the temporal hidden representation to generate the shape estimation. As described in FIG. 7, the shape decoder can comprise a
30 shape bank, linear activations, rectifiers, and a Softmax regression layer, wherein the shape bank stores shapes for estimating the estimated interaction metric.

[0102] At step 2412, the computer can generate an estimated interaction metric for a time period using the scale of the interaction data and the shape estimation of the interaction data. The computer can generate the estimated interaction metric for the time period using an amalgamate layer that uses the predicted scale of the interaction data with the shape estimation of the interaction data. The estimated interaction metric can be fed to external modules of the computer for further processing, such as a fraud detection module (e.g., the estimated interaction metric can be used for training of the fraud detection module).

VIII. COMPUTER SYSTEM

[0103] FIG. 25 shows a block diagram of an exemplary computer system 2400 according to embodiments of the present invention.

[0104] Any of the computer systems mentioned herein may utilize any suitable number of subsystems. Examples of such subsystems are shown in FIG. 25 in computer apparatus 2500. In some embodiments, a computer system includes a single computer apparatus, where the subsystems can be components of the computer apparatus. In other embodiments, a computer system can include multiple computer apparatuses, each being a subsystem, with internal components. A computer system can include desktop and laptop computers, tablets, mobile phones and other mobile devices.

[0105] The subsystems shown in FIG. 25 are interconnected via a system bus 75. Additional subsystems such as a printer 74, keyboard 78, storage device(s) 79, monitor 76, which is coupled to display adapter 82, and others are shown. Peripherals and input/output (I/O) devices, which couple to I/O controller 71, can be connected to the computer system by any number of means known in the art such as input/output (I/O) port 77 (e.g., USB, FireWire®). For example, I/O port 77 or external interface 81 (e.g. Ethernet, Wi-Fi, etc.) can be used to connect computer system 10 to a wide area network such as the Internet, a mouse input device, or a scanner. The interconnection via system bus 75 allows the central processor 73 to communicate with each subsystem and to control the execution of a plurality of instructions from system memory 72 or the storage device(s) 79 (e.g., a fixed disk, such as a hard drive, or optical disk), as well as the exchange of information between subsystems. The system memory 72 and/or the storage device(s) 79 may embody a computer readable medium. Another subsystem is a data collection device 85, such as a camera, microphone, accelerometer, and the like. Any of the data mentioned herein can be output from one component to another component and can be output to the user.

[0106] A computer system can include a plurality of the same components or subsystems, e.g., connected together by external interface 81, by an internal interface, or via removable storage devices that can be connected and removed from one component to another component. In some embodiments, computer systems, subsystem, or apparatuses can
5 communicate over a network. In such instances, one computer can be considered a client and another computer a server, where each can be part of a same computer system. A client and a server can each include multiple systems, subsystems, or components.

[0107] Aspects of embodiments can be implemented in the form of control logic using hardware circuitry (e.g. an application specific integrated circuit or field programmable gate
10 array) and/or using computer software with a generally programmable processor in a modular or integrated manner. As used herein, a processor can include a single-core processor, multi-core processor on a same integrated chip, or multiple processing units on a single circuit board or networked, as well as dedicated hardware. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will know and appreciate other ways
15 and/or methods to implement embodiments of the present invention using hardware and a combination of hardware and software.

[0108] Any of the methods described herein may be totally or partially performed with a computer system including one or more processors, which can be configured to perform the steps. Thus, embodiments can be directed to computer systems configured to perform the
20 steps of any of the methods described herein, potentially with different components performing a respective step or a respective group of steps. Although presented as numbered steps, steps of methods herein can be performed at a same time or at different times or in a different order. Additionally, portions of these steps may be used with portions of other steps from other methods. Also, all or portions of a step may be optional. Additionally, any of the
25 steps of any of the methods can be performed with modules, units, circuits, or other means of a system for performing these steps.

[0109] The specific details of particular embodiments may be combined in any suitable manner without departing from the spirit and scope of embodiments of the disclosure. However, other embodiments of the disclosure may be directed to specific embodiments
30 relating to each individual aspect, or specific combinations of these individual aspects.

[0110] The above description of example embodiments of the present disclosure has been presented for the purposes of illustration and description. It is not intended to be exhaustive

or to limit the disclosure to the precise form described, and many modifications and variations are possible in light of the teaching above.

- 5 **[0111]** A recitation of "a", "an" or "the" is intended to mean "one or more" unless specifically indicated to the contrary. The use of "or" is intended to mean an "inclusive or," and not an "exclusive or" unless specifically indicated to the contrary. Reference to a "first" component does not necessarily require that a second component be provided. Moreover, reference to a "first" or a "second" component does not limit the referenced component to a particular location unless expressly stated. The term "based on" is intended to mean "based at least in part on."
- 10 **[0112]** All patents, patent applications, publications, and descriptions mentioned herein and in the appendix are incorporated by reference in their entirety for all purposes. None is admitted to be prior art. Where a conflict exists between the instant application and a reference provided herein, the instant application shall dominate.

WHAT IS CLAIMED IS:

- 1 1. A method comprising, performing by a computer:
2 receiving interaction data between electronic devices of a plurality of entities,
3 wherein the interaction data is used to form (1) an entity interaction vector containing a
4 number of interactions between the electronic devices of a chosen entity of the plurality of
5 entities and the plurality of entities and (2) an entity time series containing a plurality of
6 metrics per unit time of the interactions of the electronic devices of the chosen entity;
7 generating, using an interaction encoder, an interaction hidden representation
8 of the entity interaction vector using embeddings of the plurality of entities, wherein the
9 embeddings represent interactions between the electronic devices of the plurality of entities;
10 generating, using a temporal encoder, a temporal hidden representation of the
11 entity time series, wherein the temporal hidden representation captures time-varying patterns
12 of the interactions between the electronic devices of the plurality of entities;
13 generating, using the interaction hidden representation and the temporal
14 hidden representation, a predicted scale comprising a magnitude of a target interaction
15 metric;
16 generating, using the interaction hidden representation and the temporal
17 hidden representation, a shape estimation of the target interaction metric; and
18 generating an estimated interaction metric of a time period using the predicted
19 scale and the shape estimation.
- 1 2. The method of claim 1, further comprising:
2 providing the estimated interaction metric to an interaction database, wherein
3 the estimated interaction metric is provided to external modules of the computer.
- 1 3. The method of claim 1, wherein the interaction hidden representation
2 of the entity interaction vector is generated using an interaction encoder that uses the entity
3 interaction vector and an entity embedding matrix to generate the interaction hidden
4 representation.
- 1 4. The method of claim 3, wherein the interaction encoder comprises a
2 normalization module that operates on the entity interaction vector to generate a normalized
3 entity interaction vector and a vector-matrix multiplication module that operates on the entity

4 embedding matrix and the normalized entity interaction vector to generate the interaction
5 hidden representation.

1 5. The method of claim 1, wherein the temporal hidden representation of
2 the entity time series is generated using a temporal encoder that uses the entity time series to
3 generate the temporal hidden representation.

1 6. The method of claim 5, wherein the temporal encoder comprises a
2 plurality of residual blocks and an average pooling layer.

1 7. The method of claim 1, wherein the predicted scale further comprises
2 an offset of the target interaction metric.

1 8. The method of claim 7, wherein the predicted scale is generated using
2 a scale decoder comprising a plurality of linear activations layers and rectifiers.

1 9. The method of claim 1, wherein the plurality of metrics per unit time
2 of the interactions comprises one or more of a number of interactions per unit time, a number
3 of declined interactions per unit time, or a number of unique accounts used to perform
4 interactions per unit time.

1 10. The method of claim 1, wherein the shape estimation of the interaction
2 data is generated using a shape decoder that uses the interaction hidden representation and the
3 temporal hidden representation to generate the shape estimation.

1 11. The method of claim 10, wherein the shape decoder comprises a shape
2 bank, linear activations, rectifiers, and a Softmax regression layer.

1 12. The method of claim 11, wherein the shape bank stores shapes for
2 estimating the estimated interaction metric.

1 13. The method of claim 1, wherein the estimated interaction metric is
2 generated using an amalgamate layer that uses the predicted scale of the interaction data with
3 the shape estimation of the interaction data, and wherein the amalgamate layer is associated
4 with a loss function that minimizes a difference of the estimated interaction metric and the
5 shape estimation.

1 14. The method of claim 1 further comprising:

2 receiving, by the computer from electronic devices of a plurality of entities,
3 access requests comprising the interaction data, wherein each access request is for a
4 destination entity of the plurality of entities; and
5 updating the entity interaction vector and the entity time series associated with
6 the destination entity of the plurality of entities using the interaction data of the access
7 request.

1 15. A computer comprising:
2 a processor; and
3 a non-transitory computer readable medium comprising instructions
4 executable by the processor to perform operations including:
5 receiving interaction data between electronic devices of a plurality of entities,
6 wherein the interaction data is used to form (1) an entity interaction vector containing a
7 number of interactions between the electronic devices of a chosen entity of the plurality of
8 entities and the plurality of entities and (2) an entity time series containing a plurality of
9 metrics per unit time of the interactions of the electronic devices of the chosen entity;
10 generating an interaction hidden representation of the entity interaction vector,
11 wherein the interaction hidden representation captures information about interactions
12 between the electronic devices of the plurality of entities;
13 generating a temporal hidden representation of the entity time series, wherein
14 the temporal hidden representation captures inherent patterns of the interactions between the
15 electronic devices of the plurality of entities;
16 generating a predicted scale comprising a magnitude and an offset of a shape
17 estimation of the interaction data of the interaction data;
18 generating the shape estimation of the interaction data; and
19 generating an estimated interaction metric for a time period using the
20 predicted scale of the interaction data and the shape estimation of the interaction data.

1 16. The computer of claim 15, wherein the interaction hidden
2 representation of the entity interaction vector is generated using an interaction encoder that
3 uses the entity interaction vector and an entity embedding matrix to generate the interaction
4 hidden representation, and wherein the interaction encoder comprises a normalization module
5 that operates on the entity interaction vector to generate a normalized entity interaction vector

6 and a vector-matrix multiplication module that operates on the entity embedding matrix and
7 the normalized entity interaction vector to generate the interaction hidden representation.

1 17. The computer of claim 15, wherein the temporal hidden representation
2 of the entity time series is generated using a temporal encoder that uses the entity time series
3 to generate the temporal hidden representation, and wherein the temporal encoder comprises
4 a plurality of residual blocks and an average pooling layer.

1 18. The computer of claim 15, wherein the predicted scale of the
2 interaction data comprises a magnitude and an offset and is generated using a scale decoder
3 that uses the interaction hidden representation and the temporal hidden representation to
4 generate the scale, and wherein the scale decoder comprises a plurality of linear activations
5 and rectifiers.

1 19. The computer of claim 15, wherein the shape estimation of the
2 interaction data is generated using a shape decoder that uses the interaction hidden
3 representation and the temporal hidden representation to generate the shape estimation, and
4 wherein the shape decoder comprises a shape bank storing shapes for estimating the
5 estimated interaction metric, linear activations, rectifiers, and a Softmax regression.

1 20. The computer of claim 15, wherein the estimated interaction metric is
2 generated using an amalgamate layer that uses the predicted scale of the interaction data with
3 the shape estimation of the interaction data, and wherein the amalgamate layer is associated
4 with a loss function that minimizes a difference of the estimated interaction metric and the
5 shape estimation.