

Method And Apparatus For Predicting Occurrence Of An Event To Facilitate Asset Maintenance

FIELD

5

The present invention relates to a method and apparatus for predicting occurrence of an event to facilitate asset maintenance.

BACKGROUND

10

Many systems comprise a plurality of assets, for instance, hardware/software components that may over time be worn down, damaged, fail to operate, or become prone to data errors and corruption. An example of such systems includes a Public Train System where operation safety is very important. Hence, it is beneficial to be able to predict events to facilitate asset maintenance for such systems so that maintenance can be attended promptly to prevent any system or component failure/damage.

15

Existing predictive models capable of providing predictions for asset maintenance are simplistic, and/or limited in the parameters considered to provide good predictions for a system such as the Public Train System which may have complex relationship structure between assets or entities. The selection process for the parameters to input to the predictive models is a difficult task requiring much effort to work out. Some of the existing predictive models capable of providing predictions for asset maintenance are neural network based. However, there are some limitations and drawbacks in such neural network models.

20

25

For instance, a temporal point process has been a principled framework for modeling temporal event series data (Reference: O. Aalen, O. Borgan, and H. Gjessing. 2008. Survival and Event History Analysis: A Process Point of View. Hereinafter "[1]"). One of the most important factors in modeling a temporal point process is the design of its conditional intensity function $\lambda(t)$. Traditionally, the temporal point process assumes that its conditional intensity has particularly specified parametric forms.

30

For example, as the simplest point process, a Poisson process (Reference: J. F. C. Kingman. 1993. Poisson processes. Vol. 3. Oxford University Press. Hereinafter "[11]") assumes its intensity function to be independent on history data. Both the homogeneous Poisson process ($\lambda(t) = \lambda_0 \geq 0$) and its time-varying generalization ($\lambda(t) = g(t) \geq 0$) share this assumption. A Hawkes process (Reference: ALAN G. HAWKES. 1971. Spectra of some self-exciting and mutually exciting point processes. Biometrika 58, 1 (1971), 83–90. Hereinafter "[7]"), on the other hand, assumes that the occurrence of each historical event increases the intensity by a certain amount. Its intensity function is defined as $\lambda(t) = \gamma_0 + \alpha \sum_{t_i < t} \gamma(t, t_i)$, where $\gamma_0 \geq 0$ is a baseline intensity independent on the history and $\gamma(t, t_i)$ is a triggering kernel capturing temporal dependencies. In contrast to Hawkes process, a Self-correcting process (Reference: Valerie Isham and Mark Westcott. 1979. A self-correcting point process. Advances in Applied Probability 37 (1979), 629–646. Hereinafter "[9]") adopts a concept that the occurrence probability of new events decreases if an event has occurred recently. The intensity function is defined as $\lambda(t) = \exp(\mu t - \sum_{t_i < t} \alpha)$, where $\mu > 0, \alpha > 0$. That is, the intensity is decreased by multiplying a constant $e^{-\alpha} < 1$ every time a new event happens. An Autoregressive Conditional Duration process (Reference: Robert Engle and Jeffrey R. Russell. 1998. Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data. Econometrica 66, 5 (1998), 1127–1162. Hereinafter "[6]") tries to capture the dependency between inter-event durations. Its intensity function is $\lambda(t) = \phi_N^{-1}(t)$, where $\phi_i = \gamma_0 + \sum_{j=0}^m \alpha_j d_{i-j}$ captures the influences from the most recent m durations, and N(t) is the total number of events up to time t.

40

45

50

One major limitation of the above traditional temporal point process methods is that they have to assume a fixed parametric form for their intensity functions in advance. However, such kind of hypothesis of the underlying event dynamics model is hard to specify or verify in practice. Thus the expressive power of these models may be restricted.

In view of the limitations of the traditional temporal point process, a few attempts have been made to design a more flexible model which can be automatically adapted to the data. The intuition is to use the recurrent neural network (RNN) to learn a general and effective representation of the underlying dynamics from the event history, so as to get rid of the parametric assumption.

Nan Du et al (Reference: Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In KDD. 1555–1564. Hereinafter “[5]”) is the first to propose the concept of the recurrent temporal point process. They proposed an RMTTP model which embeds the timing and mark information to parameterize the generation process of the observed temporal event sequences. The advantage of RMTTP is twofold. Firstly, RMTTP automatically learns the influences from the event history, hence no parametric assumption has to be made. Secondly, RMTTP directly incorporates marker information in its intensity function. Although effective, RMTTP has some limitations. On the one hand, a study on Intensity RNN (Reference: Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M. Chu. 2017. Modeling the Intensity Function of Point Process Via Recurrent Neural Networks. In AAAI. 1597–1603. Hereinafter “[18]”) observed that RMTTP can only sample transient time series features when an event happens and use a partially parametric form for the base intensity. Hence, it proposes to use a time series RNN to timely track the spontaneous background and an event sequence RNN to capture the long-range dependency over history with arbitrary time intervals. The time series RNN has its units aligned with time series indexes, while the event sequence RNN has its units aligned with asynchronous events. On the other hand, a study on CYAN-RNN (Reference: Shenghua Liu Jinhua Gao Xueqi Cheng Yongqing Wang, Huawei Shen. 2017. Cascade Dynamics Modeling with Attention-based Recurrent Neural Network. In IJCAI. 2985–2991. Hereinafter “[19]”) found that RMTTP employs a chain structure to capture the historical effect when modeling the intensity function of the temporal point process. However, in some scenarios, the events may not always happen sequentially i.e. an event could be triggered by its non-immediate predecessor in the memory chain. CYAN-RNN proposes an attention-based RNN to capture such kind of cross-dependence in the event sequence, and further introduces a coverage strategy to combat the misallocation of attention caused by the memoryless of traditional attention mechanism.

In contrast to the above two lines of researches, Hongyuan Mei et al (Reference: Hongyuan Mei and Jason Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In NIPS. Hereinafter “[12]”) proposes a neural Hawkes process which lies in the middle of them. Specifically, it designs a neurally self-modulating multivariate point process, whose intensity function is conditioned on the hidden state learnt from a novel continuous-time Long Short-Term Memory (LSTM). The update of its cell gate takes a parametric form of a Hawkes process. Such a generative model allows past events to influence the future in complex and realistic ways, by conditioning future event intensities on the hidden state of a recurrent neural network that has consumed the stream of past events.

Most of existing recurrent temporal point process work, for instance, [5], [18], and [12] models the events sequentially. Although CYAN-RNN [19] proposes to use attention mechanism to capture the cross-dependence in the event sequence, it does not distinguish multiple relation types in the dependency structure such as ontology structure of entities.

Moreover, the RNN model of CYAN-RNN [19] still relies on the temporal sequential event input i.e. the hidden state of the $(i+1)$ -th event depends on all the previous i events. However, in some cases, multiple events may occur at the same time. Hence, CYAN-RNN [19] did not consider such kind of temporal structure in the event occurrence.

5

SUMMARY

In accordance with one aspect of an example of the present disclosure, there is provided an apparatus for predicting occurrence of an event to facilitate asset maintenance, the apparatus comprising: a memory; and a processor for executing instructions in the memory to operate the apparatus to: receive data of a maintenance log comprising: a plurality of asset identifiers, each asset identifier identifying an asset; a plurality of event markers, each event marker indicating an event that occurred to each asset identified by the corresponding asset identifier; and a plurality of timestamps, each timestamp indicating duration of each event that occurred; determine an asset ontology vector for a target asset, the asset ontology vector comprising: a target asset functionality vector comprising number of occurrences of the events occurring to the target asset; and an ontology location vector of the target asset comprising values of distances between an asset identifier of the target asset and asset identifiers of other assets, the ontology location vector being derived from a hierarchy defined for the plurality of asset identifiers, wherein each asset identifier is a node in the hierarchy and distance between every pair of nodes in the hierarchy is indication of a level of similarity between each pair of nodes; extract features from the data of the maintenance log to form a vector defining values of an operation time window and a feature vector for the target asset, the feature vector comprising: a vector comprising values of number of the events that occurred to the target asset under each event category; a vector comprising values of number of the events that occurred to all assets under each category; a vector comprising values of number of the assets identified by the plurality of asset identifiers; a vector comprising values of number of all events in the data of the maintenance log; a vector comprising values of sum of timestamp for all events in the data of the maintenance log; a vector comprising values of mean of timestamp for all events in the data of the maintenance log; a vector comprising values of variance of timestamp for all events in the data of the maintenance log; a vector comprising values of a latest event occurring to the target asset and occurrence time of the latest event; generating equations of a neural network using the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset for machine learning; and output from the neural network a predicted event that will occur and time of occurrence of the predicted event.

The data of a maintenance log received may further comprises a plurality of event categories, each event marker being categorized under one of the plurality of event categories and the vector of the feature vector comprising values of number of the events that occurred to the target asset comprises values of number of the events that occurred to the target asset under each event category, the vector of the feature vector comprising values of number of the events that occurred to all assets comprises values of number of the events that occurred to all assets under each category, the vector of the feature vector comprising values of number of all events comprises values of number of the events under each category, the vector of the feature vector comprising values of number of all events in the data of the maintenance log comprises values of number of the events under each category, a vector of the feature vector comprising values of sum of timestamp for all events in the data of the maintenance log comprises values of sum of timestamp for the events under each category, a vector of the feature vector comprising values of mean of timestamp for all events in the data of the maintenance log comprises values of mean of timestamp for the events under each category, a vector of the feature vector comprising values of variance of

timestamp for all events in the data of the maintenance log comprises values of variance of timestamp for the events under each category, and the vector of the feature vector comprising values of the latest event occurring to the target asset and occurrence time of the latest event comprises values of an event category of the latest event occurring to the target asset.

The apparatus may be operated to: group the assets based on level of similarity of the asset ontology vectors calculated for each asset; calculate probability distribution of each asset over all groups of asset; and assign each asset to a group with highest probability, wherein the group of the target asset is inputted with the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset into equations of an RNN-Based neural network for machine learning.

The asset ontology vector may include geospatial information about geographical location of deployment of the target asset.

The data of the maintenance log may comprise a plurality of text description for each event that occurred to each asset identified by the corresponding asset identifier and the apparatus is further operated to: generate one or more further sub-categories for each event category from the plurality of text description, wherein the feature vector further comprises: a vector comprising values of sub-categories under each category; a vector comprising values of sum of timestamp for all events under each sub-category with state changes; a vector comprising values of mean of timestamp for all events under each sub-category with state changes; and a vector comprising values of variance of timestamp for all events under each sub-category with state changes.

The neural network may have a model that is Recurrent Neural Network (RNN) based and trains using a recurrent marked temporal point process.

The apparatus may be further operated to: generate an event vector (x_i) for each event marker based on the feature vector and the asset identifier of the target asset, generate a time vector (f_i) based on the vector defining values of an operation time window and timestamp of each event; and generate a relationship matrix (R) for each event marker based on the asset ontology vector.

The apparatus may be further operated to: determine an aggregated hidden state equation for all event markers indicated in each relationship matrix for each event marker; apply a first attention layer of an attention mechanism to learn a first weight for each event marker of each event marker; and apply a second attention layer of the attention mechanism over the output of the first attention layer to learn a second weight.

The apparatus may be further operated to: input the event vector (x_i), the time vector (f_i), the first weight, the second weight, and the aggregated hidden state equation into an update gate equation, a reset gate equation and a hidden state equation of a Gated Recurrent Unit Recurrent Neural Network (GRU RNN) to obtain a modified update gate equation, a modified reset gate equation and a modified hidden state equation and conduct the machine learning using the modified update gate equation, the modified reset gate equation and the modified hidden state equation.

The apparatus may be further operated to: apply a softmax function derived using the modified hidden state equation to determine probability of occurrence of the event indicated by each event marker.

The apparatus may be further operated to: apply a conditional intensity function and a conditional density function derived using the modified hidden state equation to determine estimated time of occurrence of the event indicated by each event marker.

In accordance with another aspect of an example of the present disclosure, there is provided a method for predicting occurrence of an event to facilitate asset maintenance, the method comprising: receiving data of a maintenance log comprising: a plurality of asset identifiers, each asset identifier identifying an asset; a plurality of event markers, each event marker indicating an event that occurred to each asset identified by the corresponding asset identifier; and a plurality of timestamps, each timestamp indicating duration of each event that occurred; determining an asset ontology vector for a target asset, the asset ontology vector comprising: a target asset functionality vector comprising number of occurrences of the events occurring to the target asset; and an ontology location vector of the target asset comprising values of distances between an asset identifier of the target asset and asset identifiers of other assets, the ontology location vector being derived from a hierarchy defined for the plurality of asset identifiers, wherein each asset identifier is a node in the hierarchy and distance between every pair of nodes in the hierarchy is indication of a level of similarity between each pair of nodes; extracting features from the data of the maintenance log to form a vector defining values of an operation time window and a feature vector for the target asset, the feature vector comprising: a vector comprising values of number of the events that occurred to the target asset under each event category; a vector comprising values of number of the events that occurred to all assets under each category; a vector comprising values of number of the assets identified by the plurality of asset identifiers; a vector comprising values of number of all events in the data of the maintenance log; a vector comprising values of sum of timestamp for all events in the data of the maintenance log; a vector comprising values of mean of timestamp for all events in the data of the maintenance log; a vector comprising values of variance of timestamp for all events in the data of the maintenance log; a vector comprising values of a latest event occurring to the target asset and occurrence time of the latest event; generating equations of a neural network using the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset for machine learning; and outputting from the neural network a predicted event that will occur and time of occurrence of the predicted event.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will be better understood and readily apparent to one skilled in the art from the following written description, by way of example only and in conjunction with the drawings, in which:

Figure 1 is a block diagram illustrating modules of an example of the present disclosure.

Figure 2 is a block diagram illustrating operation of a category refiner module of an example of the present disclosure.

Figure 3 is a block diagram illustrating derivation of an asset ontology vector by an asset ontology analyzer of an example of the present disclosure.

Figure 4 is a block diagram illustrating inputs and outputs of an RNN-Based predictor module of an example of the present disclosure.

Figure 5 illustrates existing predictive models (i.e. Fig. 5(a) to (d)) and a predictive model proposed for an example of the present disclosure.

5 DETAILED DESCRIPTION

An example of the present disclosure includes an apparatus designed to predict time and type of anomaly happening to components of a system in future. The anomaly that may occur in future may be an event that that would lead to failure of certain components of the system. The prediction of the time of occurrence of the anomaly and the prediction of the anomaly to occur performed by the apparatus helps to identify which component causing the anomaly would require maintenance such as repairs or replacements.

The apparatus includes a Category Refiner 102, an Asset Ontology Analyzer 104, an Ontology-Based Asset Cluster 106, a Feature Extractor 108, and an RNN-Based Predictor 110 (RNN refers to Recurrent Neural Network), as shown in Figure 1. Data of parameters of a Maintenance log 112 are inputted to the Category Refiner 102 and an Asset Ontology Analyzer 104. The Asset Ontology Analyzer 104 provide inputs to the Ontology-Based Asset Cluster 106 and the Feature Extractor 108 extracts features output from the Asset Ontology Analyzer 104 and the Category Refiner 102. The Ontology-Based Asset Cluster 106 and the Feature Extractor 108 then provide inputs to the RNN-Based Predictor 110.

The Maintenance log 112 of the present example includes Automatic Train Supervision (ATS) logs. A typical ATS log has categories $\{Cat_1, Cat_2, \dots, Cat_m\}$. For example, in a Mass Rapid Transit (i.e. a Public Train; abbreviated as "MRT" in the present disclosure) system, the ATS log categories include ICONIS, Trigger, Activation, InActivation, and Others which categorize the logs in high and coarse level. In the ATS log, the Activation events and InActivation events correspond to the beginning and the end of an alarm which is an anomaly in the metro or train system. Thus the *anomaly* is referred to as *alarm* in the present disclosure. Each event log E , basically includes an involved asset ID a (system component), a timestamp t , a category Cat , and a textual description e about the event. Events in certain categories involve state changes (i.e. begin and end) and therefore include an attribute of time duration d . For the ease of definition, each event is defined as $\log E = (a, t, Cat, e, d)$. For the events involving state changes, the corresponding $d \geq 0$. For the events without state changes, the corresponding d is always 0. In particular, the event description e consists of English words and symbols. Although it is described that category is being considered, it is appreciated that in some maintenance/diagnostic logs, there may not be a parameter or field for category. In this case, either the category is omitted as a parameter for machine learning that is performed later, or categories are worked out from the text description of the events that occurred.

With reference to Figure 2, when predicting the occurrence time and the type of the alarm happening to a target asset (i.e. system component), the Category Refiner 102 firstly performs pre-processing operations on the textual description e 202 about an event in the inputted historical ATS logs (i.e. the Maintenance log 112 of Figure 1) in order to provide finer category information and to reduce the dimensionality of the e 202. Since the logs are automatically generated by the ATS system, the text content of the description is constructed based on certain pre-defined description templates and variables related to the system components. Thus, the description template is extracted from event description in data pre-

processing. Each description template is referred to as a *Description Type*, denoted as *DT*. Each DT is used as a refined category for an event. To extract the DTs, the variables in the event description are identified by the following rules:

- (1) *Station names/IDs* 204 are typical variables. These acronyms are identified in all event description. For example, in the ATS logs of the MRT system, the “DDD” in the asset ID “CDDD” is an acronym of a station name, as shown in Table 1 below. For example, “CDBG” refers to Dhoby Ghuat station, which is the name of a train station in Singapore.
- (2) *Special words* 206 solely consisting of capitalized letters/digits and/or certain special symbols are usually indicating the identifiers of the components in the metro or train system. For example, “PV29” is the identifier of a train.
- (3) *Words in brackets* 208 are usually related to the status or the identifiers of the components in the metro or train system.

Table 1: Example of the ATS logs in the MRT system.

Asset ID	TimeStamp	Category	Description	Duration
CDBG/SIG/CBI/SZ1300_1	1-5-2016 1:23:28.771	Activation	Secured Zone A&B Request SZ1300_1 Electrical Section Shut Down : Active- unack	0:00:00.000
CDBG/SIG/CBI/SZ1300_1	1-5-2016 4:04:34.730	InActivation	Secured Zone A&B Request SZ1300_1 Electrical Section Shut Down : Inactive-unack	2:41:05.959
CDBG/SIG/CBI/SB0326_CA	9-11-2016 3:45:16.496	ICONIS	Subroute SB0326_CA Status Cleared	0:00:00.000

- Once the variables, *Station names/IDs* 204, *Special words* 206 and *Words in brackets* 208 are identified, a step 210 where each letter is replaced with a symbol “*” and each digit is replaced with a symbol “\$” in these variables is carried out. With the variable extraction and replacement, the event description with the same description template but different variables falls in the same description type. A benefit of the step 210 of aggregating the event description is to reduce the sparsity of the dataset and to improve the reliability of the correlation analysis. The pre-processing step produces a set of refined category *DTs* 212 for each category *Cat*, denoted as $\{DT_1, DT_2, \dots, DT_k\}_{Cat}$. Therefore, each event in the ATS logs is converted to $E = (a, t, Cat, DT, d)$.

- It is noted that the Category Refiner 102 is an optional feature and refining of categories may not be required if the input maintenance/diagnostic log 112 contains textual description e about events with sufficient dimensionality. Category refinement can be done when there is a limited number of categories in the maintenance log 112 and the description about events indicate several variables under each category that is also associated with the occurrence of the event. In this case, each category can then be divided into a plurality of sub-categories i.e. the plurality of *DTs* 212 that is each associated to the occurrence of the event. Each sub-category of a category is a *DT*. In this manner, dimensionality is increased and more

scenarios leading to the occurrence of the event are considered. The predicted results from the RNN-Based predictor 110 with inputs based on such refined categorization would thus be more accurate.

- 5 Meanwhile, with reference to Figure 3, the Asset Ontology Analyzer 104 calculates an asset ontology vector 314 for a target asset with Asset ID 302 based on the asset functionality 308, ontology location 310, and geospatial location 312 extracted from the ATS logs.

10 In particular, the target asset functionality 308 is modelled as a vector $F = (\theta_{DT_1}, \theta_{DT_2}, \dots, \theta_{DT_n})$ where θ_{DT_i} ($i = 1, 2, \dots, n$) is the number of the occurrences of the events with DT_i happening to the target asset in the historical logs. This is because the asset functionality 308 determines the operations in the system and corresponding log events associated with the operations. Ontology location 310 of the target asset is extracted from a system ontology hierarchy based on the asset IDs in the logs. The system ontology hierarchy indicates relationship between assets, wherein each asset identifier (i.e. identifier of an asset) is a node in the hierarchy and distance between every pair of nodes in the hierarchy is indication of a level of similarity between each pair of nodes. A typical asset ID 302 in ATS logs contains hierarchical information about the metro or train system. For example, the asset ID CDBG/SIG/CBI/SZ1300_1 includes the ancestor assets of the asset SZ1300_1 and the corresponding tree-like hierarchy. Therefore, the system hierarchical structure can be extracted from the asset IDs in the ATS logs and form a station asset tree and a train asset tree. The ontology location 310 of the target asset with Asset ID 302 is defined as a vector $O = (o_1, o_2, \dots, o_m)$ where o_i ($i = 1, 2, \dots, m$) is the ontology distance between the target asset and the other asset in the corresponding tree. The assets which are closer to each other in terms of ontology distance are more likely to have higher similarity. Besides the functionality and ontology location, the other part of the ontology vector includes geospatial information 312 about the asset which indicates where the target asset is physically deployed. In ATS logs of the present example described, the geo-location 312 of the station assets can be extracted from the asset ID 302, such as CDBG/SIG/CBI/SZ1300_1 where DBG is the name of a Dhoby Ghaut station ("Dhoby Ghaut" is the name of a place in Singapore). The assets in physically adjacent stations often collaborate on certain operations such as scheduling the departure and arrival time of a running train. The geo-location is encoded as a numerical identifier G according to the train line map. Thus the ontology vector of the target asset consists of F , O , and G . The category 304 in Figure 3 refers to existing categories relating to the target asset that is extractable from the ATS logs and the refined category DT 212 refers to the output of the Category Refiner 102 as described earlier.

40 Although geospatial information 312 about the asset is considered in the present example, it is an optional feature and geospatial information need not be considered for maintenance logs that has no regard to geographical location of the assets, for example, the situation where all assets are located in one location.

45 Since the metro or train system includes huge number of assets with various functionalities and different ontology location and deployed at different geo-location, simply establishing a single prediction model for all assets regardless of their differences can cause over-characterization issues and harm the prediction accuracy, while setting up a unique prediction model for each asset can suffer from data sparsity and computation workload problems. Therefore, the Ontology-Based Asset Cluster module 106 of Figure 1 groups the assets based on the similarity of the asset ontology vectors. In particular, as described in the previous paragraph, for an asset a_i , the Asset Ontology Analyzer module 104 of Figure 1 extracts an asset ontology vector 314 containing the functionality vector F_i 308, the ontology location vector O_i 310, and the geo-location G_i 312. Based on O_i , the ontology distance

between a_i and any other asset a_j can be calculated. The ontology distance is denoted as $d_{ij}^{(o)} = o_{ij}$ where $o_{ij} \in O_i$. If the asset a_i is a station asset, the geospatial distance between a_i and any other station asset a_j can be calculated. With reference to Figure 4, the asset a_i is denoted as Asset A 402, which is one of the inputs to the RNN-Based Predictor 110. The

geospatial distance is denoted as $d_{ij}^{(g)} = |G_i - G_j|$. In order to group the assets with similar functionality vector and smaller ontology distance and geospatial distance, the ontology-driven asset clustering can be model as the mathematical optimization problem as $\min_{C,Z} |F - C \cdot Z|_F^2 + \sum_k^K \sum \frac{\lambda^{(k)}}{d_{ij}^{(k)}} |\vec{C}_i - \vec{C}_j|^2$ where C and Z are the resulting matrices of the

factorization of F and $K = \{o, g\}$. $\lambda^{(k)}$ is the weighting parameter. At last, C for each asset is estimated and represents the probability distribution over M groups for the asset. The asset can be assigned to the group with the highest probability. After grouping of the asset, with reference to Figure 4, such asset group G 404 forms one of the inputs to the RNN-Based Predictor 110 of Figure 1.

As mentioned above, the asset grouping feature is to prevent over-characterization issues and any impact to the prediction accuracy that may occur due to the present example of a metro or train system, which includes huge number of assets with various functionalities and different ontology location and deployed at different geo-location. However, in simpler systems with lesser number of assets, lesser asset functionalities, less complex ontology location to consider, and/or less or no geo-location to consider, the asset grouping feature is optional.

The Feature Extractor module 108 of Figure 1 models the system context and extract features from the system context. For the ease of modelling the system context within a time window and reducing the computation cost, daily operating time of the metro or train system is divided into time buckets with an equal time interval. Each time window contains a fixed number of consecutive time buckets. In particular, assuming the length of the daily operating time is L and the time interval is set to s , there would be $\lfloor L/s \rfloor$ time buckets every day. Each time bucket B_i range from time $t_{i \cdot s}$ to $t_{(i+1) \cdot s}$. Each time window covers w consecutive time buckets, which is denoted as $W = \{B_i, B_{i+1}, \dots, B_{i+w-1}\}$. Since the time window slides over the time bucket every day, there are $\lfloor L/s \rfloor - w + 1$ time windows each day.

At the current time t_R , there is the system context C within the time window $W = \{B_c, B_{c+1}, \dots, B_{c+w-1}\}$ ranging from time $t_{c \cdot s}$ to $t_{(c+w) \cdot s}$ where the time $t_{(c+w) \cdot s}$ is the is prior to and closest to the time t_R (i.e. $t_R \in [t_{(c+w) \cdot s}, t_{(c+w+1) \cdot s}]$ falls in the time bucket B_{c+w}). Six types of features to model each system context C within the time window W are defined. With reference to table 2 below, the detailed definition of the features is as follows:

(1) Target asset a related features form a vector which consists of the total number of the events under each categories associated to the asset a , because the target alarm could be caused by the other events under different categories associated with a .

(2) Time related context feature is a non-negative integer indicating the time window of the system context because the target alarm may occur periodically in a day. A typical time related feature is the index of the last time bucket included in the system context which indicates the end of the time window of the system context. For example, given the time window $W = \{B_c, B_{c+1}, \dots, B_{c+w-1}\}$, this feature value is $c + w - 1$.

(3) Asset-DT related context features is a vector includes the total number of the events associated with station assets and the total number of the events associated with train assets under each category. An asset in metro or train system either belongs to a station or belongs to a train. A typical operation often involves multiple station assets and train assets and generates the corresponding events under different categories. For instance, this type of features includes the total number of the events under ICONIS category which are associated with any assets of stations of MRT circle line (CCL; The circle line refers to a particular train line or track in Singapore). Thus these features are extracted because certain target alarms could be caused by those relevant events associated with stations or trains. In a case where there is just one type of assets and no separate type of assets like the station assets and train assets in example described above, the vector includes the total number of the events associated with just the one type of assets under each category. However, in the case there is more than one type of assets, the vector includes the total number of the events associated with all the types of assets under each category.

(4) Asset related context features form a vector indicating the total number of the distinct assets of stations and the total number of the distinct assets of trains which are involved in the system context. Since operations can involve multiple assets at the same time to fulfil the tasks involved in the system context, the number of the assets involved in the operation may change if the target alarm occurs during the operation. Thus this type of features can examine correlation between the number of assets involved in the system context and the target alarm. In a case where there is just one type of assets and no separate type of assets like the station assets and train assets in example described above, the vector includes the total number of the distinct assets associated with just the one type of assets which are involved in the system context. However, in the case there is more than one type of assets, the vector includes the total number of the distinct assets of all the types of assets which are involved in the system context.

(5) DT related context features include the total number of events and the total number of distinct event DTs under each category. In addition, the values of sum, mean, and variance of the duration for the events are also calculated as the features. For each category, the number of the occurrences of the events with each distinct description type DT is another feature. For example, the number of the occurrences of each event description type under ICONIS is included in this type of feature. Since the events under InActivation is involved and have non-negative duration, the sum, mean, and variance of the duration of the events under InActivation associated with each distinct description types are further included. These features are used to examine how the occurrences of the events under different categories in the system context affect the occurrences of the target alarms since these events could be correlated with the target alarms in the same or relevant operations.

(6) Latest alarm related context features include the type DT of the latest alarm happening to the target asset a and the occurrence time of the alarm.

Table 2: Extracted features

Feature Type	Feature Name
(1) Target asset a related features	The number of the events involving a in each category. If there are m categories, there will be m numbers as features.
(2) Time	The time window of the system context, such as the index of the last time

related	bucket in the system context which indicates the end of the time window.
(3) Asset-DT related context features	The number of the events involving station assets in each category.
	The number of the events involving train assets in each category.
(4) Asset related context features	The number of the distinct assets of stations appearing in the system context.
	The number of the distinct assets of trains appearing in the system context.
(5) DT related context features	The number of events in each category.
	The number of the distinct DTs in each category. This is applicable in the case where each category is divided into sub-categories and each sub-category is described by a distinct DT different from other sub-categories.
	Sum of all event duration
	Mean of all event duration
	Variance of all event duration
	For each DT in a category with state changes (if any), calculate Sum of all events with description template as DT. This is applicable in the case where each category is divided into sub-categories and each sub-category is described by a distinct DT different from other sub-categories.
	For each DT in a category with state changes (if any), calculate Mean of all events with description template as DT. This is applicable in the case where each category is divided into sub-categories and each sub-category is described by a distinct DT different from other sub-categories.
	For each DT in a category with state changes (if any), calculate Variance of all events with description template as DT. This is applicable in the case where each category is divided into sub-categories and each sub-category is described by a distinct DT different from other sub-categories.
(6) Latest alarm related	For each DT in a category, calculate the number of the events with description template as DT. This is applicable in the case where each category is divided into sub-categories and each sub-category is described by a distinct DT different from other sub-categories.
	The type DT (i.e. description) of the latest alarm (or event) happening to the target asset a
	The occurrence time of the latest alarm (or event) happening to the target asset a

All of the above six types of features to model each system context C within the time window W that are defined is known collectively as the feature vector 406 in Figure 4, which is one of the inputs to the RNN-based predictor 110 of Figure 1.

5

With reference to Figure 4, based on the group of the target asset i.e. asset group G 404 and the system context feature vector i.e. feature vector 406 calculated by the Ontology-Based Asset Cluster module 106 and the Feature Extractor module 108 respectively for a particular Asset A 402, the RNN-Based Predictor module 110 selects the prediction model for the group and takes the system context feature vector as input to estimate the occurrence time 410 (i.e. occurrence time of the coming alarm or event 408) and the type of the coming alarm (or event) 408 happening to the target asset A 402 in the future.

10

Information on the predictive model of the RNN-Based Predictor 110 used for the present example is described as follows. The proposed model is herein known by a few terms in the present disclosure, which refer to the same thing. The terms are Multi-Relation Structure RNN based Recurrent Marked Temporal Point Process (MRS-RMTPP) model and multi-relation structure RNN model. Although one main MRS-RMTPP model is proposed in the present disclosure, it is appreciated that other suitable models may also be used.

Event sequences with marker and timing information are available in a wide range of domains, such as Automatic Train Supervision (ATS) system log, social network information cascades, etc. The interval between the occurrence time of two successive alarms or events and the relations among them carry important information about the dynamics of the event generation process. Correctly capturing such dynamics help to predict what type of alarm or event (i.e. 408 in Figure 4) will happen next and when (i.e. occurrence time 410 of Figure 4), which is of great importance to a lot of applications such as MRT or any other types of maintenance service, product advertising, etc. Temporal point process is one powerful tool to solve such problem and has been widely used in existing methods. Especially, the recent recurrent temporal point process methods utilize RNN to get rid of the parametric form assumption for their conditional density functions. Although effective, current recurrent temporal point process methods overlook the relation structure and temporal structure in the event sequences. In the present disclosure, a multi-relation structure RNN model is designed to capture the above two structures and propose a recurrent temporal point process algorithm with its density function conditioned on the event sequence embedding learnt from the multi-relation structure RNN model. The proposed MRS-RMTPP model outperforms the state-of-the-art baseline by 5.3% (MRR) and 4.3% (RMSE), on a real-world ATS log dataset.

Event sequence with marker information is becoming increasingly available in a broad range of domains, such as Automatic Train Supervision (ATS) system log, electronic health records, information diffusion in social networks, and so on. Each sequence contains the information about both the event marker (e.g., type, participator) and timing (i.e., when the event occurs). One popular problem about such kind of marked temporal event sequences is to answer the question "*what type of event will happen next and when*". This problem is of great importance in various domains. For example, in ATS system, to predict when a MRT train will break down and what type of error causes the breakdown can facilitate the MRT maintenance service and help to provide a smooth travel experience to the passengers. And in the information diffusion scenario, predicting who will be the next to diffuse this piece of information and when the diffusion will happen can be useful for product advertising, news spreading, etc.

A lot of efforts have been devoted to solving this problem. Among them, temporal point process is a powerful tool to model such temporal event sequences. The early attempts usually make a hypothesis of the rules that govern the generative process of the events, based on which a fixed parametric form is defined for their conditional density function so as to approximate the probability when the next event will happen. The representative studies include Poisson process [11], Hawkes process [7], Self-correcting process [9], Autoregressive Conditional Duration process [6], and so on. Recently, some researchers observed that the performance of such kind of methods highly depend on the whether the hypothesis reflects the reality, which requires strong prior knowledge of the sequence data. To address this limitation, recent studies start explore recurrent neural network (RNN) in their recurrent temporal point process methods [5], [18], [19]. Specifically, they use RNN to embed the sequential events with both markers and timing, and then condition their density function on the learnt embedding. By utilizing RNN to capture the sequential dependence or effects in the event sequence, the recurrent temporal point process now avoids the parametric form assumption.

Although effective, most of recurrent temporal point process methods model the events sequentially. That is, the embedding of a node depends on the embedding of the previous node in the event sequence. This sequential process has two major limitations. Firstly, they neglect the relation structure in the event sequences. Take Figure. 5 as an example, most of the existing methods/techniques process the events sequentially (e.g. Fig. 5(a) i.e. sequence 502). CYAN-RNN [19] notices that there may exist cross-dependence in the event sequence (e.g. Fig. 1(b) i.e. sequence 504). That is, an event may be triggered by its non-immediate predecessor in the event chain. For example, the occurrence of e_3 depends on e_1 only rather than e_1 and e_2 . CYAN-RNN [19] proposes to use the attention mechanism to capture such cross-dependence in the event sequence. However, it does not distinguish multiple relation types in the dependency structure. For example, in Fig. 5(c) i.e. sequence 506, the dependency of e_4 on e_2 and e_4 are of different types. How to utilize this different types of relation dependency to more accurately model the event dynamics is a problem. Secondly, CYAN-RNN [19] overlooks the temporal structure in the event sequences. Multiple events may occur at the same time (e.g., Fig. 5(d) i.e. sequence 508). This also breaks the sequential dependency of events. I.e., e_2 and e_3 happen at the same time, so e_3 depends on e_1 only, even though e_2 may be the predecessor of e_3 in the memory chain. By comprehensively considering both the relation structure and the temporal structure in the event sequences, the event sequence is modelled as illustrated in Fig. 5(e) i.e. sequence 510. For each event, there is leveraging on the embedding from all the events that "happen" before it, and differentiating of different types of relation dependencies. E.g., e_5 has only one "child" dependency from e_1 . Fig. 5(e) i.e. sequence 510 is an illustration of the multi-relation structure RNN model that is proposed herein.

To model the relation structure and temporal structure in event sequences, a novel multi-relation structure RNN model is designed. A recurrent marked temporal point process is proposed whose density function is conditioned on the embedding learnt by the multi-relation structure RNN model. Specifically, the Multi-Relation Structure RNN based Recurrent Marked Temporal Point Process model (MRS-RMTPP) takes the event sequence with marker, timing and ontology relations as the input. At each timestamp, the past events sequence is represented as a history embedding, which embeds the information of marker and timing for each event, as well as the relations among them. A marked temporal point process is then trained by condition of its density function on the learnt embeddings. The advantages of the MRS-RMTPP model are two-fold. Firstly, the MRS-RMTPP model frees the temporal point process from any parametric form assumption. Secondly, the MRS-RMTPP model comprehensively embeds all the information (marker, timing and relation ontology) into one powerful and expressive representation, leading to the performance improvements over current recurrent temporal point process.

Major contributions of the present disclosure are summarized as follows.

- The relation structure and the temporal structure in the event sequences is identified, which are two important factors when modeling the temporal point process.
- A novel multi-relation structure RNN is designed to properly model the above two structures in the event sequences. It distinguishes the multiple types of relations and properly processes the co-occurrence of events.
- A recurrent temporal point process algorithm with the density function conditioned on the multi-relation structure RNN model is proposed. The experiments described in the present disclosure verify the effectiveness of the algorithm over the state-of-the-art baselines.

Input data may be a list of discrete temporal events $\{y_i, t_i, e_i\}$, with the marker $y_i \in \mathcal{Y}$, the timing $t_i \in \mathbb{R}^+$, the entity $e_i \in \mathcal{E}$ and $i \in \mathbb{Z}^+$. A tuple $\{y_i, t_i, e_i\}$ denotes that an event y_i happen on entity e_i at time t_i . Note that $t_i \leq t_{i+1}$. That is to say, multiple events may happen at the same time. The entities have inherent ontology structure. Let $r(e_i, e_j)$ denote the relation

between entity e_i and entity e_j . E.g., $r(e_i, e_j) = \text{child}$ denotes that e_i is a child of e_j . Denote $\mathcal{R}$ as the set of relation types, i.e., $r \in \mathcal{R}$. The event marker and entity may have different meanings in different application scenarios. For example, in Automatic Train Supervision (ATS), the event sequences correspond to the ATS log streams. An example of an entity-event pair is "(signal system, not response)". Different components in the train system have the ontology structure. In information diffusion, the event sequences can be the cascades (i.e., sequences of re-sharing behaviors ascendingly ordered by time). In this case, the event marker is the user who diffused the information and the entity is the user herself. The ontology of entities is the friendship relations among users. An aim of the present disclosure is to build a general model which is able to:

- Predict the next event-time pair (y_{i+1}, t_{i+1}) given a sequence of the past events;
- Evaluate the likelihood of a given sequence of events;
- Simulate a new sequence of events based on the learnt parameters of the MRS-RMTPP model.

Two preliminary concepts i.e. Gated Recurrent Unit and Marked Temporal Point Process are first discussed in sections 1.1 and 1.2 as follows.

1.1 Gated Recurrent Unit

Gated Recurrent Unit (GRU) (Reference: KyungHyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. CoRR abs/1409.1259 (2014). Hereinafter "[3]") is one variant of the recurrent neural network (RNN). RNN handles the variable-length sequence by having a recurrent hidden state whose activation at each time is dependent on that of the previous time (Reference: Junyoung Chung, Çağlar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR abs/1412.3555 (2014). Hereinafter "[4]"). It has been observed that RNN cannot capture long-term dependencies because the gradients tend to either vanish or explode (Reference: Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning Long-term Dependencies with Gradient Descent is Difficult. Trans. Neur. Netw. 5, 2 (March 1994), 157–166. Hereinafter "[2]"). GRU is then designed to address this issue by making each recurrent unit to adaptively capture dependencies of different time scales.

Formally, given a sequence $x = (x_1, x_2, \dots, x_T)$ as the input, at each timestamp t , GRU updates its recurrent hidden state (or the activation) h_t by a linear interpolation between the previous activation h_{t-1} and the candidate activation $\hat{h}_t$:

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \hat{h}_t \quad (1)$$

where $\circ$ is an element-wise multiplication. An update gate z_t decides how much the unit updates its activation and is calculated by:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. The above procedure takes a linear sum between the existing state and the newly computed state. The candidate activation $\hat{h}_t$ is computed similarly to that of the traditional recurrent unit as follows:

$$\hat{h}_t = \tanh(W_h x_t + U_h (r_t \circ h_{t-1}) + b_h) \quad (3)$$

where $\tanh(x) = 2/(1 + e^{-2x}) - 1$ is the hyperbolic tangent function and r_t is a reset gate calculated by:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (4)$$

1.2 Marked Temporal Point Process

Marked temporal point process is a powerful tool to model the latent mechanisms governing the observed random event patterns along time [5]. Formally, the input data is a list of

discrete temporal events $\{t_i, y_i\}$, with the timing $t_i \in \mathbb{R}^+$, the marker $y_i \in \mathcal{Y}$ and $i \in \mathbb{Z}^+$. Let the history $\mathcal{H}_t = \{t_i, y_i | t_i < t\}$ denote the list of event time and marker pairs up to the time t .

- 5 Given the history of past event time pairs $\mathcal{H}_t$, marked temporal point process capture the dynamics of the temporal events by specifying the conditional density function that the next event will happen at time t with type y as $f(t, y | \mathcal{H}_t)$. Then the joint likelihood of observing a sequence is derived as:

$$f(\{(t_i, y_i)\}_{i=1}^N) = \prod_i f(t_i, y_i | \mathcal{H}_{t_i}) \quad (5)$$

- 10 Due to the excessive complications caused by jointly and explicitly modeling the timing and marker information [5], the common assumption is that the event type and event time are conditionally independent with each other, and the factorization of the above equation is :

$$f(t_i, y_i | \mathcal{H}_{t_i}) = f(y_i) f(t_i | \dots, t_{i-2}, t_{i-1}) = f(y_i) f(t_i | \mathcal{H}_{t_i}) \quad (6)$$

- 15 where f_{y_i} is a multinomial distribution of the event types for the next event. $f(t_i | \mathcal{H}_{t_i})$ is the conditional density function which denotes the likelihood that the next event will occur at time t_i given the timing sequence of the past events.

In the temporal point process, the conditional density function $f(t | \mathcal{H}_{t_i})$ is modeled as:

$$f(t | \mathcal{H}_{t_i}) = \lambda(t | \mathcal{H}_{t_i}) \exp \left(- \int_{t_{i-1}}^t \lambda(\tau | \mathcal{H}_{t_i}) d\tau \right) \quad (7)$$

- 20 where $\lambda(t | \mathcal{H}_{t_i})$ is the conditional intensity function which denotes the probability of a new event (i.e., y_i) happening within a small window $[t_{i-1}, t]$.

Consequently, the next event time is predicted using the expectation of Eq. 7 as follows:

$$t_i^* = \int_{t_{i-1}}^{\infty} t \cdot f(t | \mathcal{H}_{t_i}) dt \quad (8)$$

- 25 Based on different parametric assumptions about how the observed point patterns are generated, different forms of the conditional intensity function $\lambda(t)$ have been designed (Reference is made to [1]). Representative examples includes Poisson process ($\lambda(t) = g(t) \geq 0$) [11], Hawkes process ($\lambda(t) = \gamma_0 + \alpha \sum_{t_i < t} \gamma(t, t_i)$) [7], Self-correcting process ($\lambda(t) = \exp(\mu t - \sum_{t_i < t} \alpha)$) [9], Autoregressive Conditional Duration process ($\lambda(t) = \phi_{N(t)}^{-1}, \phi_i = \gamma_0 + \sum_{j=0}^m \alpha_j d_{i-j}$) [6], etc.
- 30

- 35 The above methods 1.1 and 1.2 share one common limitation, i.e., one has to make an assumption of a fixed parametric form for the observed temporal data series. In view of this, RNN has recently been explored to learn the representation of the underlying latent dynamics from the history data so that no assumption has to be made in advance.

In the sections 2.1 to 2.3 as follows, the details of the proposed model design and how to conduct an end-to-end training is provided.

40 2.1 Multi-Relation Structure RNN

- An intuition applied in the present disclosure is that the probability of an event happening increases if any of its related events has been observed. The event relations are reflected in the entity ontology structure. Take ATS log streams as an example, the related entities include its parent entities, children entities, sibling entities. The affects of different types of relations and even the affects of different entities within one relation type may be different.
- 45

RNN is adopted to learn a general representation to approximate the unknown model that governs the event occurrences in the history data. The reason is that RNN has a feedback mechanism which creates an internal state of the network to memorize the influence of each pas data sample [5]. That is, in Eq. 1, the hidden state at current timestamp h_t not only depends by current input $\hat{h}_t(x_t)$, but also depends by the hidden state from the last timestamp h_{t-1} . Similarly, h_{t-1} is updated by considering both h_{t-2} and x_{t-1} . In this way, the hidden state at time t embeds the influence of the historical events up to time t . Recall that $\mathcal{H}_t = \{t_i, y_i | t_i < t\}$ denotes the list of event time and marker pairs up to the time t . It is easy to get that the hidden state h_{t-1} in RNN represents $\mathcal{H}_t$. Then the conditional density function $f(t_i | \mathcal{H}_t)$ in Eq. 6 can be naturally converted to:

$$f(t_i | \mathcal{H}_t) = f(t_i | h_{t-1}) \quad (9)$$

By embedding the historical event marker and timing information as a vector, it is now able to design a more general conditional intensity function $\lambda(t)$ without having to specify a parametric form for the dependency structure over the history. An objective is then to design an RNN model which is capable of modeling the event sequences with multi-relation structure and temporal structure.

To properly model the multi-relation ontology in the event sequences, in the present example, a structure GRU with a two-layer attention mechanism is proposed. It is noted that more than 2 layer attention mechanism or other suitable/similar techniques may also be used. Structural RNN has been explored in a few studies, such as

- tree-RNN (Reference: Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In ACL. 1556–1566. Hereinafter “[16]”),
- DAG-RNN (Reference: Bing Shuai, Zhen Zuo, BingWang, and GangWang. 2016. DAG-Recurrent Neural Networks for Scene Labeling. In CVPR. 3620–3629. Hereinafter “[15]”),
- topological-RNN (Reference: Jia Wang, Vincent W. Zheng, Zemin Liu, and Kevin Chen-Chuan Chang. 2017. Topological Recurrent Neural Network for Diffusion Prediction. In ICDM. Hereinafter “[17]”), etc.

However, none of the above methods have considered the multiple types of relationships in the ontology. Attention mechanism has been widely used in various deep learning models to automatically learn the pooling weights from the observed data. However, use of attention mechanism for the purposes of predicting occurrence of an event to facilitate asset maintenance is never done before. To leverage the inferences from different relations more accurately, a two-layer attention mechanism is constructed for an example of the present disclosure. The first attention layer is constructed above the hidden units within one type of relation so as to differentiate the effects from different predecessors of the same relation type. The second attention layer is constructed above the aggregated hidden state of different relations (i.e., the outputs of the first layer attention) to differentiate the contributions of each relation.

To deal with the co-occurrence of multiple events at one timestamp, the input of the multi-relation structure RNN model is changed from a sequence to a temporal structure. At each timestamp, each of the event tuples is individually processed and only the related events happen before the current timestamp are considered. The hidden state of each event tuple is derived as follows.

At time t , for each tuple $\{y_i, t_i, e_i\}$ where $t_i = t$, a vector representation of it is learnt using an embedding layer by $x_i = W^{(e)}y_i + b^{(e)}$, where e_i is a one-hot vector representation of the event marker. In addition to this dynamic embedding learnt from the multi-relation structure RNN

model, a 134-d feature f_i is also extracted based on the timestamp of the event (t_i). f_i contains information about "month of the year", "day of the month", "day of the week", "hour of the day", "minute of the hour". Both x_i and f_i are inputs into the multi-relation structure RNN model. Linking back to the earlier disclosure, f_i is generated based on (2) Time Related Feature Type in table 2. x_i is generated based on (1) Target asset a related features, (3) Asset-DT related context features, (4) Asset related context features, (5) DT related context features, and (6) Latest alarm related in table 2.

For a tuple $\{y_i, t_i, e_i\}$, let $R(i)$ denote the entities who have relationship R with entity e_i and have been observed in the sequence. For example, $Parent(i)$ denotes e_i 's parents set who happen before t_i . In the case of ATS log, $R \in \{Parent, Children, Sibling\}$.

Then for each single relation type $R \in \mathcal{R}$, the update gate $z_i^{(R)}$, reset gate $r_i^{(R)}$ and candidate activation $\hat{h}_i^{(R)}$ are learnt as follows:

$$z_i^{(R)} = \sigma(W_z^{(x)} x_i + W_z^{(f)} f_i + U_z^{(R)} \bar{h}_i^{(R)} + b_z) \quad (10)$$

$$r_i^{(R)} = \sigma(W_r^{(x)} x_i + W_r^{(f)} f_i + U_r^{(R)} \bar{h}_i^{(R)} + b_r) \quad (11)$$

$$\hat{h}_i^{(R)} = \tanh(W_h^{(x)} x_i + W_h^{(f)} f_i + U_h^{(R)} (r_i^{(R)} \circ \bar{h}_i^{(R)}) + b_h) \quad (12)$$

where $\bar{h}_i^{(R)} = \sum_{v_j \in R(i)} \alpha_j^{(R)} h_j$ is the aggregated hidden state of e_i 's predecessors who has relation R with e_i .

The first layer attention weight $\alpha_j^{(R)}$ are learnt for each single type of relation. These entity attention weights are learnt to distinguish the influences from different entities of the same type. For each relation type $R \in \mathcal{R}$ and for each entity $e_j \in R(i)$, its entity attention weights $\alpha_j^{(R)}$ are learnt as follows:

$$\beta_j^{(R)} = \eta_j^{(R)} \cdot \tanh(Q^{(R)} h_j + b^{(R)}) \quad (13)$$

$$\alpha_j^{(R)} = \frac{\exp(\beta_j^{(R)})}{\sum_{v_l \in R(i)} \exp(\beta_l^{(R)})} \quad (14)$$

The final hidden state for the tuple $\{y_i, t_i, e_i\}$ is then calculated as:

$$h_i = \sum_{R \in \mathcal{R}} \tilde{\alpha}_R \left\{ (1 - z_i^{(R)}) \circ \bar{h}_i^{(R)} + z_i^{(R)} \circ \hat{h}_i^{(R)} \right\} \quad (15)$$

The second layer attention weights $\tilde{\alpha}_R$ differentiate the contributions from different relations. This $|\mathcal{R}|$ -dimensional relation attention $\tilde{\alpha}$ is learnt as follows:

$$\tilde{\beta}_R = \tilde{\eta} \cdot \tanh(\tilde{Q} \left\{ (1 - z_i^{(R)}) \circ \bar{h}_i^{(R)} + z_i^{(R)} \circ \hat{h}_i^{(R)} \right\} + \tilde{b}) \quad (16)$$

$$\tilde{\alpha}_R = \frac{\exp(\tilde{\beta}_R)}{\sum_{R \in \mathcal{R}} \exp(\tilde{\beta}_R)} \quad (17)$$

where $R \in \mathcal{R}$.

2.2 MRS-RMTPP

With the above novel multi-relation structure RNN model ready, the resultant Multi-Relation Structure RNN based Recurrent Marked Temporal Point Process model (MRS-RMTPP) is used for next event marker prediction and time prediction.

- 5 Based on Eq. 6, to model the probability of observing a tuple $\{y_i, t_i, e_i\}$, both the probability of both observing a specific event marker $f(y_i)$ and the conditional density function $f(t_i|\mathcal{H}_{t_i})$ need to be calculated.

- 10 For the event marker modeling, the multinomial distribution of the event makers for the next event using the hidden state learnt from the multi-relation structure RNN model is derived. This is realized by the below softmax function:

$$p(y_i = k|h_{i-1}) = \frac{\exp(\sum_{R \in \mathcal{R}} V_k^{y(R)} \tilde{h}_i^{(R)} + b_k^y)}{\sum_{k=1}^K \exp(\sum_{R \in \mathcal{R}} V_k^{y(R)} \tilde{h}_i^{(R)} + b_k^y)} \quad (18)$$

- 15 For the event timing modeling, as has been introduced in Eq. 9, the density function of MRS-RMTPP model is conditioned based on the history embedding learnt from the MRS-RMTPP model. The conditional intensity function of the MRS-RMTPP model is calculated as:

$$\begin{aligned} \lambda(t|\mathcal{H}_{t_i}) &= \lambda(t|h_{i-1}) \\ &= \exp \left(\sum_{R \in \mathcal{R}} V^{(R)} \tilde{h}_i^{(R)} + w^t(t - t_{i-1}) + b^t \right) \end{aligned} \quad (19)$$

where $\tilde{h}_i^{(R)} = \sum_{v_j \in R(i)} \alpha^{(R)} h_j$ is the aggregated hidden state of e_i 's predecessors who has relation R with e_i .

- 20 The above intensity function leverages three influence factors to compute the probability of next event's occurrence, including the past history events, the current event, and a background intensity score. The first term $\sum_{R \in \mathcal{R}} V^{(R)} \tilde{h}_i^{(R)}$ in Eq. 19 embeds the accumulative influence from the past related event marker and timing information. The second term $w^t(t - t_{i-1})$ encodes the effect of current event t_{i-1} . The last term provides a background intensity value for the next event appearing.

According to Eq. 7, the conditional density function of the MRS-RMTPP model is then derived as:

$$\begin{aligned} f(t|\mathcal{H}_{t_i}) &= \lambda(t|\mathcal{H}_{t_i}) \exp \left(- \int_{t_{i-1}}^t \lambda(\tau|\mathcal{H}_{t_i}) d\tau \right) \\ &= \exp \left\{ \sum_{R \in \mathcal{R}} V^{(R)} \tilde{h}_i^{(R)} + w^t(t - t_{i-1}) + b^t \right. \\ &\quad + \frac{1}{w^t} \exp \left(\sum_{R \in \mathcal{R}} V^{(R)} \tilde{h}_i^{(R)} + b^t \right) \\ &\quad \left. - \frac{1}{w^t} \exp \left(\sum_{R \in \mathcal{R}} V^{(R)} \tilde{h}_i^{(R)} + w^t(t - t_{i-1}) + b^t \right) \right\} \quad (20) \end{aligned}$$

30

The timing for the next event y_i is then estimated as:

$$t_i^* = \int_{t_{i-1}}^{\infty} t \cdot f(t|\mathcal{H}_{t_i}) dt \quad (21)$$

Note that the integration above equation does not have analytic solution, processing is followed (Reference is made to [5]) by applying numerical integration techniques (Reference: W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. 1992. Numerical Recipes in C. Cambridge University Press, Cambridge. Hereinafter "[13]") for one-dimensional functions to compute Eq. 21 instead.

2.3 End-to-End Learning

Given a set of sequences $\mathcal{S} = \{S_k\}$, where S_k is a list of discrete temporal events $\{y_i, t_i, e_i\}$, where $i = 0, 1, \dots, N_k$, the MRS-RMTPP model is trained to maximize the below joint log-likelihood of observing $\mathcal{S}$.

$$\begin{aligned} \ell(\{S_k\}) &= \log \left(\prod_k f(\{(t_i, y_i)\}_{i=1}^{N_k}) \right) = \log \left(\prod_k \prod_i f(t_i, y_i) \right) \\ &= \log \left(\prod_k \prod_i f(t_i, y_i | \mathcal{H}_{t_i}) \right) \\ &= \log \left(\prod_k \prod_i f(y_i | h_{t_{i-1}}) f(t_i | h_{t_{i-1}}) \right) \end{aligned} \quad (22)$$

$$= \sum_k \sum_i \left(\log f(y_i | h_{t_{i-1}}) + \log f(t_i | h_{t_{i-1}}) \right) \quad (23)$$

where $f(y_i | h_{t_{i-1}})$ and $f(t_i | h_{t_{i-1}})$ are formulated in Eq. 18 and Eq. 20 respectively.

The model parameters are learnt by maximizing the above log-likelihood. back-propagation through time (BPTT) is adopted for training. At every iteration, for every event in the sequence, an event embedding is firstly learnt, which together with a 134-d time feature extracted from the timestamp, is input into the multi-relation structure GRU model. The learnt event sequence embedding is then used to model the conditional density function of the recurrent marked temporal point process. Stochastic gradient descent (SGD) is applied with mini-batch and the model parameters are updated by techniques taught in Adam (Reference: Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980 (2014). Hereinafter "[10]").

In the sections 3.1 to 3.4 as follows, comparison is made between the proposed MRS-RMTPP model and the state-of-the-art baseline in a large-scale real-world ATS log dataset.

3.1 Dataset

A set of ATS log data is collected from the MRT system in Singapore. The dataset contains the ATS log for 64 circle line trains during August 1st to September 15th in 2016. The event logs for one train in one day is formed as one event sequence, and the sequence with length equals to one is removed. Finally, 2,413 sequences are derived in total. More statistics about the dataset are described in Table 3. All event sequences are sorted by the date they happened, and then select the last 20% of sequences for testing. The rest of the 80% of event sequences as used for training.

Table 3: Dataset Statistics

# sequences	# event markers	# assets	# timestamps
2,413	34	27	45,268
avg. seq. length	max. seq. length	Relations	
25.685	536	{parent, child, sibling}	

3.2 Baseline

Comparison is made between the MRS-RMTPP model and a state-of-the-art baseline TransitionP, which predicts the timing and marker of the event based on a transition matrix. More specifically, an event-event transition matrix is constructed where each element denotes the probability of a new event appearing given an observed event. Meanwhile, an event-time transition matrix is constructed to compute the probability of the occurrence time of an event when given the marker of this event. Both the sequence of event markers and the sequence of event timing information are used to train their model. TransitionP does not model the relation ontology structure among the events.

3.3 Evaluation Metrics

Performance in terms of both event marker prediction and event timing prediction is evaluated. For the evaluation of event marker prediction, Mean Reciprocal Rank (MRR) (Reference: Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating Web-based Question Answering Systems. In LREC. Hereinafter "[14]") is adopted. MRR measures whether the correct answer is ranked higher than the wrong answers in the returned ranked list. Specifically, MRR is the average of the reciprocal ranks of results for a sample of queries, where the reciprocal rank of a query is the multiplicative inverse of the rank of the first correct answer. The formal definition is:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (24)$$

where $rank_i$ denotes the rank position of the first relevant answer for the i -th query, and $|Q|$ is the total number of queries. The larger MRR is, the better it is.

In terms of the event timing prediction evaluation, Root Mean Square Error (RMSE) (Reference: Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. International Journal of Forecasting (2006), 679–688. Hereinafter "[8]") is used between the estimated time t_i^* and the groundtruth t_i . Formally,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{|Q|} (t_i^* - t_i)^2}{|Q|}} \quad (25)$$

The smaller RMSE is, the better it is.

Two variants of MRR and RMSE, i.e., macro MRR, macro RMSE and micro MRR, micro RMSE are calculated. Macro MRR(RMSE) is the average of MRR(RMSE) scores with respect to each MRT train, and micro MRR(RMSE) is the overall MRR(RMSE) with respect to all trains.

3.4 Comparison with Baselines

Comparison is made between the proposed MRS-RMTPP model and the baseline TransitionP in terms of the prediction of next event marker and timing. The results are reported in Table 4. The best results are highlighted in bold.

Table 4: Evaluation of Event Marker and Timing Prediction

Methods	Macro MRR	Micro MRR	Macro RMSE	Micro RMSE
TransitionP	0.843	0.807	1.737	1.559
MRS-RMTPP	0.849	0.850	1.849	1.495

From Table 4, it can be seen that, because of comprehensive consideration of the relation ontology structure in event sequences, the MRS-RMTPP model outperforms the baseline in most cases, especially in terms of micro evaluation scores. More specifically, the MRS-RMTPP model is consistently better than the baseline in terms of next event marker prediction. MRS-RMTPP outperforms the baseline by 0.7% in terms of Macro MRR and 5.3% in terms of Micro MRR. For next event timing prediction, the MRS-RMTPP method improves the baseline in terms of Micro RMSE by 4.3%. Although TransitionP shows better result on Macro RMSE, it only performs better for the trains with fewer event sequences (as indicated by the Micro RMSE). Moreover, the MRS-RMTPP model is significantly better than the baseline w.r.t. the micro MRR(RMSE). The better micro metric values indicate that more events are correctly predicted in terms of overall performance, which is in need and beneficial for real-life scenarios.

In the present disclosure, in one example, a novel recurrent marked temporal process to model the event sequences with both relation structure and temporal structure is proposed. The novel multi-relation structure RNN model is designed to embed the dynamics that govern the generation process of the events, and the learnt event sequence embedding is then used to model the density function of the temporal point process. Due to its capability of capturing the two structures in the event sequences, the proposed MRS-RMTPP model outperforms the state-of-the-art baseline by 5.3% (micro MRR) and 4.3% (micro RMSE), on a real-world ATS log dataset.

Throughout this specification and claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or group of integers or steps but not the exclusion of any other integer or group of integers.

While the invention has been described in the present disclosure in connection with a number of embodiments and implementations, the invention is not so limited but covers various obvious modifications and equivalent arrangements, which fall within the purview of the appended claims. Although features of the invention are expressed in certain combinations among the claims, it is contemplated that these features can be arranged in any combination and order.

CLAIMS

1. An apparatus for predicting occurrence of an event to facilitate asset maintenance, the apparatus comprising:

 - a memory; and
 - a processor for executing instructions in the memory to operate the apparatus to: receive data of a maintenance log comprising:
 - a plurality of asset identifiers, each asset identifier identifying an asset;
 - a plurality of event markers, each event marker indicating an event that occurred to each asset identified by the corresponding asset identifier;
 - and
 - a plurality of timestamps, each timestamp indicating duration of each event that occurred;
 - determine an asset ontology vector for a target asset, the asset ontology vector comprising:
 - a target asset functionality vector comprising number of occurrences of the events occurring to the target asset; and
 - an ontology location vector of the target asset comprising values of distances between an asset identifier of the target asset and asset identifiers of other assets, the ontology location vector being derived from a hierarchy defined for the plurality of asset identifiers, wherein each asset identifier is a node in the hierarchy and distance between every pair of nodes in the hierarchy is indication of a level of similarity between each pair of nodes;
 - extract features from the data of the maintenance log to form a vector defining values of an operation time window and a feature vector for the target asset, the feature vector comprising:
 - a vector comprising values of number of the events that occurred to the target asset under each event category;
 - a vector comprising values of number of the events that occurred to all assets under each category;
 - a vector comprising values of number of the assets identified by the plurality of asset identifiers;
 - a vector comprising values of number of all events in the data of the maintenance log;
 - a vector comprising values of sum of timestamp for all events in the data of the maintenance log;
 - a vector comprising values of mean of timestamp for all events in the data of the maintenance log;
 - a vector comprising values of variance of timestamp for all events in the data of the maintenance log;
 - a vector comprising values of a latest event occurring to the target asset and occurrence time of the latest event;
 - generating equations of a neural network using the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset for machine learning; and
 - output from the neural network a predicted event that will occur and time of occurrence of the predicted event.
2. The apparatus as claimed in claim 1, wherein the data of a maintenance log received further comprises a plurality of event categories, each event marker being categorized under one of the plurality of event categories and

the vector of the feature vector comprising values of number of the events that occurred to the target asset comprises values of number of the events that occurred to the target asset under each event category,

the vector of the feature vector comprising values of number of the events that occurred to all assets comprises values of number of the events that occurred to all assets under each category,

the vector of the feature vector comprising values of number of all events comprises values of number of the events under each category,

the vector of the feature vector comprising values of number of all events in the data of the maintenance log comprises values of number of the events under each category,

a vector of the feature vector comprising values of sum of timestamp for all events in the data of the maintenance log comprises values of sum of timestamp for the events under each category,

a vector of the feature vector comprising values of mean of timestamp for all events in the data of the maintenance log comprises values of mean of timestamp for the events under each category,

a vector of the feature vector comprising values of variance of timestamp for all events in the data of the maintenance log comprises values of variance of timestamp for the events under each category, and

the vector of the feature vector comprising values of the latest event occurring to the target asset and occurrence time of the latest event comprises values of an event category of the latest event occurring to the target asset.

3. The apparatus as claimed in claim 1 or 2, wherein the apparatus is operated to: group the assets based on level of similarity of the asset ontology vectors calculated for each asset;

calculate probability distribution of each asset over all groups of asset; and assign each asset to a group with highest probability, wherein the group of the target asset is inputted with the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset into equations of an RNN-Based neural network for machine learning.

4. The apparatus as claimed in claim 1, 2 or 3, wherein the asset ontology vector includes geospatial information about geographical location of deployment of the target asset.

5. The apparatus as claimed in any one of the preceding claims, wherein the data of the maintenance log comprises a plurality of text description for each event that occurred to each asset identified by the corresponding asset identifier and the apparatus is further operated to:

generate one or more further sub-categories for each event category from the plurality of text description, wherein

the feature vector further comprises:

a vector comprising values of sub-categories under each category;

a vector comprising values of sum of timestamp for all events under each sub-category with state changes;

a vector comprising values of mean of timestamp for all events under each sub-category with state changes; and

a vector comprising values of variance of timestamp for all events under each sub-category with state changes.

6. The apparatus as claimed in any one of the preceding claims, wherein the neural network has a model that is Recurrent Neural Network (RNN) based and trains using a recurrent marked temporal point process.

7. The apparatus as claimed in any one of the preceding claims, wherein the apparatus is further operated to:

generate an event vector (x_i) for each event marker based on the feature vector and the asset identifier of the target asset,

generate a time vector (f_i) based on the vector defining values of an operation time window and timestamp of each event; and

generate a relationship matrix (R) for each event marker based on the asset ontology vector.

8. The apparatus as claimed in claim 7, wherein the apparatus is further operated to:

determine an aggregated hidden state equation for all event markers indicated in each relationship matrix for each event marker;

apply a first attention layer of an attention mechanism to learn a first weight for each event marker of each event marker; and

apply a second attention layer of the attention mechanism over the output of the first attention layer to learn a second weight.

9. The apparatus as claimed in claim 8, wherein the apparatus is further operated to:

input the event vector (x_i), the time vector (f_i), the first weight, the second weight, and the aggregated hidden state equation into an update gate equation, a reset gate equation and a hidden state equation of a Gated Recurrent Unit Recurrent Neural Network (GRU RNN) to obtain a modified update gate equation, a modified reset gate equation and a modified hidden state equation and

conduct the machine learning using the modified update gate equation, the modified reset gate equation and the modified hidden state equation.

10. The apparatus as claimed in claim 9, wherein the apparatus is further operated to:

apply a softmax function derived using the modified hidden state equation to determine probability of occurrence of the event indicated by each event marker.

11. The apparatus as claimed in claim 10, wherein the apparatus is further operated to:

apply a conditional intensity function and a conditional density function derived using the modified hidden state equation to determine estimated time of occurrence of the event indicated by each event marker.

12. A method for predicting occurrence of an event to facilitate asset maintenance, the method comprising:

receiving data of a maintenance log comprising:

a plurality of asset identifiers, each asset identifier identifying an asset;

a plurality of event markers, each event marker indicating an event

that occurred to each asset identified by the corresponding asset identifier;

and

a plurality of timestamps, each timestamp indicating duration of each event that occurred;

determining an asset ontology vector for a target asset, the asset ontology vector comprising:

a target asset functionality vector comprising number of occurrences of the events occurring to the target asset; and

an ontology location vector of the target asset comprising values of distances between an asset identifier of the target asset and asset identifiers of other assets, the ontology location vector being derived from a hierarchy defined for the plurality of asset identifiers, wherein each asset identifier is a node in the hierarchy and distance between every pair of nodes in the hierarchy is indication of a level of similarity between each pair of nodes;

extracting features from the data of the maintenance log to form a vector defining values of an operation time window and a feature vector for the target asset, the feature vector comprising:

a vector comprising values of number of the events that occurred to the target asset under each event category;

a vector comprising values of number of the events that occurred to all assets under each category;

a vector comprising values of number of the assets identified by the plurality of asset identifiers;

a vector comprising values of number of all events in the data of the maintenance log;

a vector comprising values of sum of timestamp for all events in the data of the maintenance log;

a vector comprising values of mean of timestamp for all events in the data of the maintenance log;

a vector comprising values of variance of timestamp for all events in the data of the maintenance log;

a vector comprising values of a latest event occurring to the target asset and occurrence time of the latest event;

generating equations of a neural network using the asset identifier of the target asset, the asset ontology vector of the target asset and the feature vector of the target asset for machine learning; and

outputting from the neural network a predicted event that will occur and time of occurrence of the predicted event.