Time-series as Background Data for Relating Medical Diagnoses Terms

Saket Gurukar
The Ohio State University
gurukar.1@osu.edu

Srikanta Bedathur
IIT Delhi
srikanta@cse.iitd.ac.in

ABSTRACT

Relating terms from different ontologies or identifying the most relevant entry in an ontology for a given term, is an important task in various settings involving the use of ontologies. Often the task of relating terms is achieved by considering the instance level matchings within the ontologies being aligned, or using an external common ontology for indirect linking, or using annotated text corpus. In this paper, we focus on a variant of this problem that occurs in relating medical diagnosis terms. We propose a novel *unsupervised approach* that exploits the availability of time-series data medical events of patients during their stay in *intensive care unit (ICU)*. Our method, called Decree, is evaluated using a large-scale real-world medical repository of ICU data including event data from laboratory test measurements to quantify the relationship strength between terms from a given ontology. We further outline how Decree can be used to assign diagnoses terms in case of unlabeled pathology as well. We show that Decree can discover better quality relationships and is more scalable than state-of-the-art time-series techniques.

CCS CONCEPTS

• **Information systems** → *Data mining*; *Ontologies*.

KEYWORDS

Time-Series, MIMIC, Dynamic Time Warp

ACM Reference Format:
Saket Gurukar and Srikanta Bedathur. 2019. Time-series as Background Data for Relating Medical Diagnoses Terms. In *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB '19), September 7–10, 2019, Niagara Falls, NY, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3307339.3342145

1 INTRODUCTION

While assigning diagnoses to an observed set of symptoms, most medical professionals follow one of the many well-established terminologies. Unfortunately, due to the availability of multiple medical diagnoses term ontologies, the same symptoms may be assigned alternative diagnoses terms taken from different ontologies. In such situations, we need a mechanism to relate the terms obtained from different ontologies. This is accomplished by a variety of methods ranging from those which use structural and lexical features in each ontology [9, 10], or by making use of an external ontology accepted as a "clearing house" for mappings.

In this paper, we use an alternate, more data-driven, approach that makes use of the *time-series* of various diagnosis measurements of a patient. We rely on the medically proven hypothesis that similar diseases/conditions are likely to present similar behavior on various (diagnosis and monitoring) measurements over time. Such an approach offers an additional advantage of being able to identify newly developing diseases and disease strains, so that an appropriate course of action can be recommended at the earliest.

Specifically, we model the relationship between diagnoses terms used by medical professionals (from potentially multiple ontologies) using *intensive care unit (ICU) medical event-data* associated with the patient. We pursue a tangential direction from earlier work which relies on the use of external annotated text corpus for addressing the problem of measuring the relatedness between terms. Instead of relying on external annotated text corpus, we propose a novel time-series-based relationship scoring method called Decree (Discovery of Event-centric Relationships between Entities) which scores relationships using *local patterns of similarities* of time-series of ICU events. We demonstrate through the experiments that time-series-based relationship scoring method is yet another effective way to relate terms using abundantly available data in the medical domain. Specifically, we conduct experiments on MIMIC-III [7] – a large publically available ICU medical data mart – to evaluate the effectiveness of our proposed method. In order to quantify the effectiveness, we compare the top-ranked relationships determined by Decree by using their distance and depth in an expert-curated International Classification of Diseases (ICD-9) —a widely used disease ontology maintained by the World Health Organization. Our results show that using time-series of various ICU lab test individually is capable of identifying more than 60% true relationships between diagnoses terms. Further, we also experiment with different techniques for time-series similarity scoring which is at the heart of our method and show that a simpler and scalable technique we propose is more effective than the popularly used dynamic time-warping (DTW) method for time-series similarity scoring.

1.1 Contributions:

The key contributions we make in this paper are three-fold:

(1) We introduce the notion of measuring the relationship strength between diagnoses terms using time-series-based analysis over associated event data.
(2) We develop a novel common-pattern-based scoring algorithm as part of our method called Decree— Discover Event Centric RElationships between Entities — and apply it for relating medical diagnoses terms.

ACM-BCB '19, September 7–10, 2019, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-6666-3/19/09…$15.00
https://doi.org/10.1145/3307339.3342145

(3) Using the LabEvents table in MIMIC-III, a large, real-world, publically available medical dataset, we show that our approaches can identify relationships between diagnoses terms with high accuracy.

The Decree method can also be used with different time-series similarity scoring algorithms such as Dynamic Time Warp (DTW) [2]. However, we show that the proposed scoring algorithm mine better quality relationships than DTW; is more scalable than DTW and does not require the global time-series normalization necessary for DTW.

1.2 Background on MIMIC-III Data:

MIMIC-III —which stands for Medical Information Mart for Intensive Care (version 3) [7]— is a large, public medical dataset. It spans more than decade and consists of anonymized health data of over 40, 000 critical care patients at Beth Israel Deaconess Medical Center. This data includes information of admitted patient's demographics, laboratory test results, vital sign measurements at bedside, procedures, notes and so on. The MIMIC-III database consists of 26 tables, of which five tables define and track patient stays in ICU, and five tables define the high-level dictionary mappings. LabEvents table records the laboratory measurements during the course of the stay of an admitted patient. An admitted patient will undergo a series of laboratory tests. These laboratory tests are not necessarily taken at regular time intervals, but the time-stamp of actual acquisition of the sample used for the test are recorded — we use this to define our time-series data. Each patient is assigned multiple diagnoses at the time of their discharge, and they are ordered by priority. For simplicity, we consider only the highest ranked diagnoses associated with an admission in our work.

2 PROBLEM FORMULATION

In this section, we formally define the problem of relating terms using the associated time-series data. We define an ***entity***, e (and $f, g, h, \ldots$), as a subject that has multiple measurements taken against it, from a set $\mathcal{M}$ of measurement types. We call each measurement type, m, as a *feature* with $m \in |\mathcal{M}|$.

Each feature m results in a ***time-series*** of its measurements for an entity e, denoted as T^e_m and individual measurements are indicated by $T^e_m[i], T^e_m[j], \ldots$, where the indexes $i, j, \ldots$, correspond to the time-stamps of these measurements. In the same vein, we write $T^e_m[s, t)$ as a shortcut for the subsequence from the time-series from timestamp s to t. An example of time-series of measurements for two entities is shown in Figure 1.

Each feature has an associated metric and we assume all measurements of a feature use the associated metric across all entities. We make no assumptions on the time-stamps in the time-series for different features and different entities. We call the collection of all time-series for an entity as its *event-centric data* collection, $\psi^e = \left\{T^e_m \mid m \in \mathcal{M}\right\}$. These concepts can be illustrated as applied in the setting of MIMIC-III data we are working with using the following example:

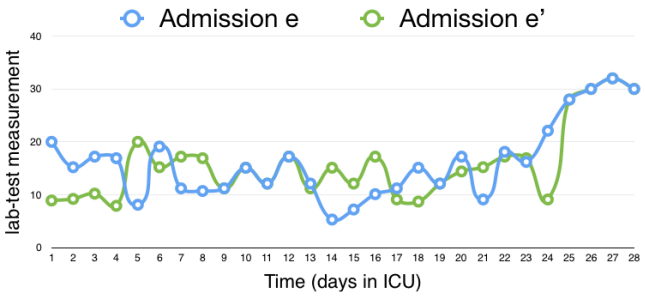


Figure 1: Time-series of a measurement for two Patient Admissions

Example 1. *An admission of a patient*[1] *e into ICU, due to severe vomiting has been diagnosed with Ischemic Stroke. Also, due to vomiting, there is an increase in pH values, measured as part of regular blood work done on patients in ICU [14]. This results in a time-series of measurements T^e_{pH} for e using which we can detect the increase in pH values.*

Since vomiting is a common indication of disorders of the central nervous system, increase in pH values can also be observed in patients with diagnoses such as Rett Syndrome (RTT) – with associated time-series $T^{e'}_{pH}$. Note that pH values show a similar pattern of increase over time until doctors administer drugs to control it. Thus, based on the pattern of variations in pH values, we can discover relationships between diagnoses terms — in this case, between Ischemic Stroke and Rett Syndrome. □

2.1 Relating Terms using Local Similarity Patterns

In this paper, for the sake of simplicity, we will work with the assumption that each entity is associated with only one diagnoses term. As we already mentioned, in MIMIC-III these terms are added at the end of the stay of a patient at ICU (i.e., at the time of their discharge), and are listed in the order of priority as determined by the health-care expert. We use the highest priority term, and treat it as the diagnoses term which has the event-centric data collection associated with the entity under consideration. Consequently, we interchangeably use T^e_m and T^d_m for the same time-series of measurements of feature m associated with an entity e with the top diagnoses term d.

The central hypothesis pursued in this paper is that the relationship strength (i.e., semantic relatedness in an ontology) between two diagnoses terms d_1 and d_2 can be determined by considering the patterns of similarity between pairs of time-series from T^{d_1} and T^{d_2} respectively. Consider the illustrative time-series data from two entities e and e' we presented earlier in Figure 1. On coarse-grain level, the two time-series appear to be dissimilar, however, fine-grain analysis reveals the patterns of *local similarity* between them.

[1]Note that we do not treat each patient as a separate entity because the same patient can be admitted into ICU at different times and may be diagnosed with different disorders.

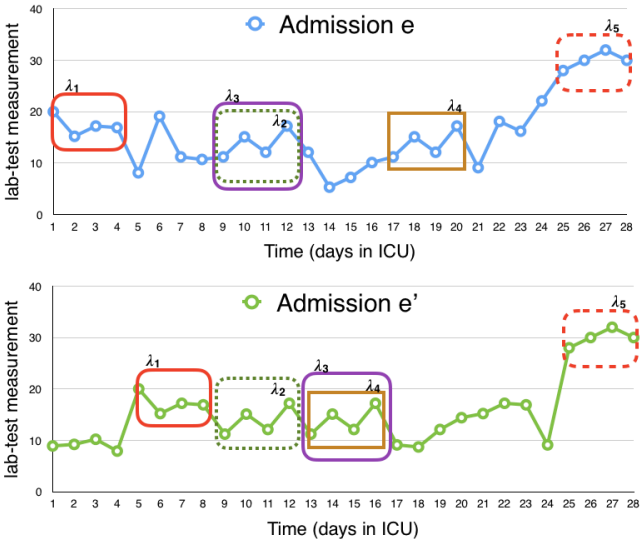


Figure 2: Different Local Similarity Patterns between Time-series Data of two Entities

Definition 1. *We say two entities e and e' have a* ***local similarity*** *on a feature m if there exists a pair of subsequences $T_m^e[s,t)$ and $T_m^{e'}[s',t')$ corresponding to entity e and e' respectively s.t. the distance – as determined by a time-series distance function D – is within a specified threshold δ. We use the notation $\lambda^{(e,e')}(s,t,s',t')$ for the local similarity pattern between entities e and e'. When the pair of entities and/or the feature are clear from context, we will drop them from the notation.* □

Figure 2 highlights the subsequences of local similarity between the two time-series data. Informally, two entities can share a local similarity on feature d if a (sub)sequence of values in a time-window of one entity are similar to a sequence of values in the same or a different time-window of the other entity.

One might consider that simple counting of all the local similarity patterns between two time-series is sufficient for computing the strength of the relationship between entities (or diagnoses terms). But there are two main problems with simple counting of all the local similarity patterns:

(1) **Presence of Noise**: A subsequence in one time-series, say, $T^e[s,t)$ starting at s may show similarity with another subsequences, $T^{e'}[s',t')$ which is separated by a large time-gap in the other time-series i.e. $t << s'$. A local similarity between $T^e[s,t)$ and $T^{e'}[s',t')$ with large time-gap might be due to coincidence or noise. The question is how much time-gap should be allowed between $T^e[s,t)$ and $T^{e'}[s',t')$ before we say the local similarity is not due to noise? The answer is dependent on the applied domain.

(2) **Overlapping patterns**: A subsequence from one time-series may show similarity in two different similarity patterns. For instance, in Figure 2, $T^e[9,13)$ is involved in similarity patterns λ_2 and λ_3. But counting them as separate patterns leads to incorrect measurements of strength. We address this by introducing the *monotonicity condition* on local similarity patterns.

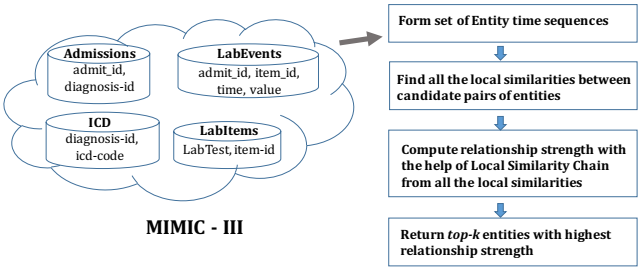


Figure 3: Overview of DECREE

Definition 2. ***Monotonicity of local similarity patterns:*** *We say two local similarity patterns $\lambda(s,t,s',t')$ and $\lambda(p,q,p',q')$ satisfy* monotonicity condition *when $t < p \wedge t' < p'$ is satisfied (Note that we already have the condition of $s < t \wedge s' < t' \wedge p < q \wedge p' < q'$ from the definition of sub-sequences). We denote two similarity patterns λ and λ' satisfying monotonicity pattern as $\lambda \ll \lambda'$.* □

Informally, the monotonicity condition states that two local similarity patterns should not *overlap* with each other. For instance, the two local similarity patterns λ_1 and λ_2 in Figure 2 are considered to satisfy monotonicity condition $\lambda_1 \ll \lambda_2$, but the patterns λ_2 and λ_3 do not i.e $\lambda_2 \not\ll \lambda_3$.

Definition 3. ***Local Similarity Chain****: Given all local similarity patterns between two entity time-series, $\lambda_1, \lambda_2, \ldots$, a* local similarity chain *is a sequence of local similarity patterns, $\lambda_l, \lambda_{l+1}, \ldots$, such that $\lambda_l \ll \lambda_{l+i}$ where $i > 0$.* □

For a given sequence of local similarity patterns between two time-series, there are *multiple* local similarity chains that can be formed between them. We define *chain length* as the number of unique local similarity patterns that are present in a local similarity chain.

Based on the concept *chain length*, we quantify the *relationship strength* between two time-series – and between the corresponding entities – as the maximum chain length over all possible chains between the two time-series.

With these definitions in place, we now can formally define the problem ***Top-k Time-series Related Entities*** as follows: *Given a set of entity time-sequences, and an integer $k \geq 1$, discover top-k pairs of entities with highest relationship strength.*

3 DECREE

The overview of the DECREE algorithm is shown in Fig. 3. A medical data lake contains information about patients, doctors, hospitals along with the information of lab tests undertaken by patients. We consider an admitted patient as an entity e and each lab test m results in a time-series of its measurements T_m^e for an entity e. Next, we find *local similarities* for each unique pair of entities and compute *relationship strength* between them with the help of *local similarity chains* formed using discovered *local similarities*. In the end, we return top-k pairs of entities with the highest *relationship strength*.

Alg. 1 presents the pseudo code of DECREE. In order to identify *local similarities*, one has to search for similar local subsequences between T_m^e and $T_m^{e'}$. The parameter *local subsequence time window* ρ is used to identify these local subsequences. We use tumbling

Algorithm 1 : DECREE (T_m , δ, ρ, ω, k)

Input: T_m time-series data for feature m encompassing all entities, δ, ρ, ω and required number of related entities k
Output: Top - k pair of entities with highest relationship strength.

1: **for each** $(T_m^e, T_m^{e'})$ in T_m.cartesian(T_m) such that $e < e'$ **do**
2: $all\_local\_similarities$ = **findLocalSimilarities** (T_m^e, $T_m^{e'}$, δ, ρ, ω)
3: $rel\_strength$ = **findRelationStrength** ($all\_local\_similarities$)
4: Add in set rel_pairs the values ($rel\_strength$, e, e')
5: $disc\_relationships$ = **takeOrdered**(rel_pairs, k) based on descending ordering on $rel\_strength$.
6: **return** $disc\_relationships$

Algorithm 2 : findLocalSimilarities (T_m^e, $T_m^{e'}$, δ, ρ, ω)

Input: T_m^e, $T_m^{e'}$, δ, ρ, ω
Output: All *local similarities* between T_m^e and $T_m^{e'}$

1: Store values of $T_m^e[s', s'+\rho)$ at index $k = (s'-s)/\rho$ where s refers to initial time of T_m^e.
2: Let I and I' be index structure for T_m^e and $T_m^{e'}$ respectively.
3: $search\_windows$ = { 0 to $\pm\ \omega/\rho$ }
4: **for each** index k in I **do**
5: $T_m^e[p, q) = I[k]$
6: **for each** index l in $search\_windows$ **do**
7: $T_m^{e'}[p', q') = I'[k + l]$
8: **if dist** $(T_m^e[p, q), T_m^{e'}[p', q'))$ < d **then**
9: Add $\lambda(p, q, p', q')$ in $all\_local\_similarities$ list
10: **return** $all\_local\_similarities$

windows to form local subsequences with the help of ρ. For instance, a local subsequence contains all values in time series $T_m^e[s, s+\rho)$. It is possible that two local subsequence $T_m^e[s, s+\rho)$ and $T_m^{e'}[t, t+\rho)$ are similar but have large time gap between them i.e. $|(t+\rho) - (s+\rho)|$ is very large. But such pairs of local subsequence might not be *correlated*. Consider an scenario where we have T_{pH}^e and $T_{pH}^{e'}$ and ρ of 1 day. It is possible that local subsequence$T_{pH}^e[1, 2)$ is similar to local subsequence$T_{pH}^{e'}[51, 52)$. Assuming both of the patients got admitted when initial symptoms of their disease were detected, initial increase of pH will be characteristic of a particular disease while increase of pH only at latter stages of patient's admission will be characteristic of another disease. Here, the similarity between the two pH local subsequences, having large time gap, was coincidental. In order to handle such scenarios, we introduce *search constraint time window* ω as another input parameter such that given $T_m^e[s, s+\rho)$ and $T_m^{e'}[t, t+\rho)$ the time gap $|(t+\rho)-(s+\rho)| \leq \omega$. The parameter ω also helps in pruning candidate local subsequences from other time-series and is responsible for reducing the running time of DECREE.

Next, we describe the steps of DECREE algorithm. The calculation of *relationship strength* between two time-series T_m^e and $T_m^{e'}$ is invariant to the order of the entities e and e', as a result the candidate pairs of entities are compared only once (Step 1). For each pair of entities, we first identify all possible *local similarities* with help of **findLocalSimilarities** function in Step 2 and in order to solve the problem of *Overlapping* patterns and *Inconsistent pattern sequences* we compute the *relationship strength* with help of function **findRelationStrength** in Step 3. Next, the pair of entities and their *relationship strength* is stored in a set rel_pairs. Once Steps 1 - 4 are executed for each unique pair of entities, the top-k pairs of entities having highest *relationship strength* are retrieved using **takeOrdered** function. DECREE algorithm can also be implemented in an distributed environment like Spark.

3.1 Discover Local Similarities

Alg. 2 presents the pseudo code for function **findLocalSimilarities**. Given a pair of Entity time sequences T_m^e and $T_m^{e'}$, *local similarity distance threshold* δ, *local subsequence time window* ρ and *search constraint time window* ω **findLocalSimilarities** function return all possible *local similarities* between T_m^e and $T_m^{e'}$. For faster retrieval of values from a time-series T_m^e, we create an index structure and all the values of $T_m^e[s', s'+\rho)$ that lies in time window $(s', s'+\rho)$ are stored at index $k = (s'-s)/\rho$ where s refers to initial time of T_m^e (Step 1). Because of indexing, the candidate local subsequences satisfying ω can be retrieved with indices from 0 to $\pm\ \omega/\rho$. Next we compare one $T_m^e[p, q)$ with another $T_m^{e'}[p', q')$ and the distance between them is less than δ, we say e and e' have a *local similarity* $\lambda(p, q, p', q')$. At last, all the local similarities between T_m^e and $T_m^{e'}$ are returned in a list.

Note that the choice of distance function is not restricted in DECREE. There are two factors one should consider before choosing distance function:

(1) **Learn Distance Function**: If time series consists multidimensional data and the features contain textual data, then in such cases, the distance between local subsequences can be calculated by learning the distance function [11].
(2) **Handle varying length subsequences**: If time series T_m^e and $T_m^{e'}$ have large disparity in length, then one can select elastic measures like dynamic time warping (DTW).

We chose Euclidean distance as distance function for both it's simplicity and effectiveness [5]. After the function **findLocalSimilarities** returns all the *local similarities* between T_m^e and $T_m^{e'}$, the next step is to calculate the relationship strength.

3.2 Relationship Strength

In order to determine the *relationship strength*, between entities using *local similarities*, one has to identify the maximum chain length from all possible *local similarity chains* (Def. 3). We define an ordering between *local similarities* in order to identify *local similarities* satisfying Monotonicity condition (Def. 2). Informally, we say $\lambda_i < \lambda_j$ when either (1) λ_i occurs before λ_j in both T_m^e and $T_m^{e'}$. For example, in Figure 2, $\lambda_1 < \lambda_2$ or, (2) In T_m^e, λ_i and λ_j occurs at same time but in $T_m^{e'}$, λ_i occurs beforeλ_j. For example, in Figure 2, $\lambda_2 < \lambda_3$.

Formally,

Definition 4. ***Local Similarity Ordering*** *: We say two local similarity* $\lambda_1(s, t, s', t')$ *and* $\lambda_2(p, q, p', q')$ *are ordered as* $\lambda_1 < \lambda_2$ *when*

(1) $t' < q'$
(2) $t' = q', s' < p'$
(3) $t' = q', s' = p', t < q$

Algorithm 3 : findRelationStrength (*all_local_ similarities*)

Input: All local similarities $all\_local\_similarities$.
Output: *Relationship strength*

1: Sort $all\_local\_similarities$ by *Local Similarity Ordering* such that $\lambda_l < \lambda_{l+k}$ where k > 0. .
2: Initialize *LocalSimilarityMap* as [local_similarity_id, $num\_local\_similarities\_statisfying\_monotonicity$]
3: $rel\_strength$ = 0, len = $all\_local\_similarities$.length
4: **for** i = len; i > 0; i -= 1 **do**
5: Get Local Similarity λ_i at index i.
6: *LocalSimilarityMap*[λ_i] = 1
7: **for** j = i + 1; j < len; j += 1 **do**
8: Get Local Similarity λ_j at index j.
9: **if** $\lambda_i \ll \lambda_j$ **then**
10: **if** *LocalSimilarityMap*[λ_i] < *LocalSimilarityMap*[λ_j] + 1 **then**
11: *LocalSimilarityMap*[λ_i] = LocalSimilarityMap[λ_j] + 1
12: **if** $rel\_strength$ < *LocalSimilarityMap*[λ_i] **then**
13: $rel\_strength$ = *LocalSimilarityMap*[λ_i]
14: **return** $rel\_strength$

(4) $t' = q', s' = p', t = q, s < p$

The algorithm to calculate *relationship strength* with the help of *local similarities* is presented in Alg. 3. Initially, we sort all the *local similarities* such that Local Similarity Ordering between two consecutive *local similarities* is maintained. Next we initialize *LocalSimilarityMap* that stores a *local similarity* λ_i and the number of *local similarities* satisfying Monotonicity condition in the Local Similarity Chain starting from λ_i. The value of λ_i in *LocalSimilarityMap* signifies the maximum *local similarity chain* length. In steps 4 - 11, we pick λ_i and check if λ_j satisfy Monotonicity condition such that $\lambda_i \ll \lambda_j$ (Step 9). If $\lambda_i \ll \lambda_j$ is true, we increment the value of maximum *local similarity chain* length of λ_j in *LocalSimilarityMap* by 1 and assign it to *LocalSimilarityMap*[λ_i]. The execution steps of Alg. 3 for *local similarities* in Fig. 2 are shown in Table 1.

Lemma 3.1. *Alg. 3 correctly calculates the* relationship strength *with the help of* local similarities.

Proof. Correctness in the calculation of the *relationship strength* requires correctness in calculation of maximum *local similarity chain* length which is represented by $rel\_strength$ in Algo. 3.

Loop invariant : *At the end of each iteration of the* ***for*** *loop of lines 4 - 13, each* local similarity λ_i *in* LocalSimilarityMap *contains maximum* local similarity chain *length in the* Local Similarity Chain *such that* $\lambda_i \ll \lambda_j$ *where* $i < j$ *and* i, j *are indices in* Local Similarity Chain

Initialization: For the *local similarity* λ_i present at index *len*, no λ_j exists and hence *Local Similarity Chain* contains only λ_i, as a result *LocalSimilarityMap[λ_i]* is set to 1. So, Loop invariant is maintained.

Maintenance: Observe that for *local similarity* λ_i, all possible candidates λ_j such that $\lambda_i < \lambda_j$ are checked with Monotonicity condition at Step 9. Once $\lambda_i \ll \lambda_j$, we pick a *local similarity* λ_j which has the maximum *local similarity chain* length. Hence Loop invariant is maintained.

Termination: At termination, $i = 0$. By the loop invariant, each *local similarity* λ_i in *LocalSimilarityMap* contains the maximum *local similarity chain* length. Since, the variable $rel\_strength$ stores the maximum *local similarity chain* length seen till now (Step 12), Algorithm 3 correctly calculates the maximum *local similarity chain* length.

Consequently, Alg. 3 correctly calculates the *relationship strength* with the help of *local similarities*.

□

Steps 4-13	Step 4 i	Step5 λ_i	Step8 λ_j	Step 12 LocalSimilarity -Map[λ_i]	Step 13 rel_strength
Iteration1	5	λ_5	-	1	1
Iteration2	4	λ_4	λ_5	2	2
Iteration3	3	λ_3	λ_4, λ_5	2	2
Iteration4	2	λ_2	$\lambda_3, \lambda_4, \lambda_5$	3	3
Iteration5	1	λ_1	$\lambda_2, \lambda_3, \lambda_4, \lambda_5$	4	4

Table 1: The value of few steps at end of *for* loop at Step 13, when Alg. 3 is executed on entities e and e' in Figure 2.

4 TASK SPECIFICATION OVER MIMIC-III

The question we ask is "Can we detect relationships between diagnosis terms by simply analysing the time series of the patient's laboratory tests?". For instance, can we automatically mine the relationship between diagnosis terms Ischemic Stoke and Rett Syndrome by analysing the patients' pH time-series T^e_{pH}.

4.1 Problem Formulation over MIMIC

Admissions table stores the information about each patient through admit_id and information about his/her diagnosis through diagnosis_id. The diagnosis_id references to the Icd table that contains the information about the diagnosis in detail. Information related to lab tests like pH, Eosinophils are stored in LabItems table with each lab test having a unique item_id. Each lab test undertaken by an admitted patient at a specific time is recorded and stored in LabEvents table. The overview of generation of time series T^e_m from MIMIC-III is shown in Figure 3. As mentioned in section 1.2, each patient is assigned multiple diagnoses – ordered by priority. For simplicity, we consider only the highest ranked diagnoses associated with admission in our work. As motivated through Example 1, relatedness between time-series based on lab tests – such as T^e_{pH} – between two patients corresponds to the relationship between their diagnoses terms with the confidence of relationship given by relationship strength (Section 3.2)

4.2 Evaluating Results using MIMIC

Now, even if we detect the relationships between diagnosis terms, how would we evaluate the quality of discovered relationships? The task to evaluate the quality is not trivial. The discovered related diagnoses can be synonyms of each other or can have an ontological relationship among themselves or the discovered diagnosis terms can have no relation. For instance, Myocardial Infarction *is-a* Ischemic Heart Disease and Intermediate Coronary Syndrome *is-a* Ischemic Heart Disease. Here both Myocardial Infarction and Intermediate Coronary Syndrome have an immediate common ancestor. This ontological relationships between diagnosis terms can

Features	Num entities	Min	Max	Mean
Atypical Lymphocytes	426	10	182	23
Bands	495	10	186	22
Basophils	959	10	205	20
Bilirubin, Direct	455	10	57	15
Bilirubin, Total	3482	10	191	22
Eosinophils	959	10	205	20
Metamyelocytes	428	10	182	23
Monocytes	959	10	205	20
Myelocytes	427	10	182	23
pH	14076	10	607	30

Table 2: Statistics of T_m^e of each feature. Num entities is the number of patients who have undertaken a laboratory test. Min, Max and Mean refers to the minimum, maximum and the average of the lengths of T_m^e for a particular feature.

be found out with the help of biomedical ontology. MIMIC - III uses the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD) diagnosis tool to assign codes to each disease. ICD also contains the ontological relationships between these ICD codes. We calculate the relationship score between two diagnosis terms using these ICD codes. We use the metric proposed by Wu and Palmer [12] that takes into consideration both the distance and depth of diagnosis terms in the hierarchy. Let LCS (D_1, D_2) be the Least Common Subsumer [12] which refers to the least common ancestor between D_1 and D_2 in the ICD hierarchy. The *True Relationship Score (TRS)* between two diagnosis terms is then defined as:

$$TRS(D_1, D_2) = \frac{2 * depth(LCS(D_1, D_2))}{depth(D_1) + depth(D_2)} \quad (1)$$

If there is no common ancestor between a pair of diagnosis, then LCS is 0, as a result, *TRS* is bounded between 0 and 1. The quality of k-pairs of diagnosis is given by *Average True Relationship Score* (*ATRS*).

5 EXPERIMENTAL SETUP AND RESULTS

In this section, we experimentally validate the idea of discovering related diagnosis terms by performing time series analysis on *event-centric data* ψ^e. All the experiments are performed on a 3-node Spark cluster with each node running on CentOS and each node having 4 cores of Intel(R) Xeon(R) CPU E7-4830 v2 @ 2.20GHz along with 14.5 GB of RAM. We use two time series analysis techniques for discovering related entities

Dynamic Time Warping (Dtw) We implemented the state-of-the-art time series analysis [5] technique Dtw on Spark. The Dtw distance between all unique pairs of entity time series T_m^e is computed. The smaller the Dtw distance between pair of entities, the higher is the strength of relationship between those entities. The top-k related entities using Dtw are k-pair of entities having smallest Dtw distance (hence have highest relationship strength).

DECREE Presented in Section 3.

To verify the idea of "Discovering related entities by performing time series analysis" generates statistically significant results, we identify related entities using Random technique. In Random, given all entities and their time series T_m^e, we randomly pick two entities and say they are related. Both the techniques Dtw and Decree returns a ranked list of related entities in descending order of relationship strength. For evaluation, we pick top - 1000 ranked relationships from both Dtw and Decree and 1000 randomly generated entity pairs from Random.

Params	Local similarity distance threshold δ	Local subsequence time window ρ	Search constraint time window ω
Decree short	0.25	0.5 days	1.5 days
Decree long	0.25	1 days	4 days

Table 3: Parameters of Decree short and Decree long.

Next, we show that for detecting relationships Decree that utilizes the concept of local similarities performs better than Dtw in terms of accuracy, ranking and speed. The features from MIMIC-III we use in the following experiments are presented in Table 2 along with their statistics about their time series T_m^e. We use two sets of parameters for Decree mainly 'Decree short' and 'Decree long' shown in Table 3 for evaluating the results in this section. An extensive study on multiple sets of parameters is presented in Section 5.5.

5.1 Ranking

Both Decree and Dtw returns a ranked list of related entities in descending order of relationship strength between entities. The ranked list of Decree and Dtw are compared using Discounted Cumulative Gain (DCG) [6]. $DCG_p = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)}$ where rel_i is relevance score which in our case is *True Relationship Score* of i^{th} rank entity pair. In Decree, the pair of entities are ranked with the help of relationship strength while in Dtw, the Dtw distance is used for ranking. If two pairs of diagnosis terms have the same relationship strength or same Dtw distance, we take the average of ranks for tied scores and compute DCG with updated ranks. In Random, the related entities list is generated randomly. Figure 4 shows the comparison of ranking between Decree, Dtw and Random using Discounted Cumulative Gain for various features. As one can see, 'Decree short' or 'Decree long' parameters keep high ranked related diagnosis terms at the top of the ranked list as compared to Dtw. In order to verify the fact that Decree keeps the strongly related entities at the top, we show the heatmap of Top-20 related entities from both Decree and Dtw in Figure 7. It is clear that Decree lists related entities with high *True Relationship Scores* at the top of the mined list as compared to Dtw.

5.2 Accuracy

To compare the accuracy of Decree, Dtw and Random against all the features m, we use *Average True Relationship Score*(*ATRS*), as defined in Section 4.2. We show the results in Figure 5. We see that 'Decree short' is better than Dtw and Random in all the features except pH, while 'Decree long' performs betters than other techniques for pH feature.

Reason for Low Accuracy of DTW in few features : In features, Metamyelocytes and Myelocytes, the *ATRS* of Dtw with Z-normalization is smaller than Random. The reason is close to 67% of Metamyelocytes values in lab tests have a value of 0, which

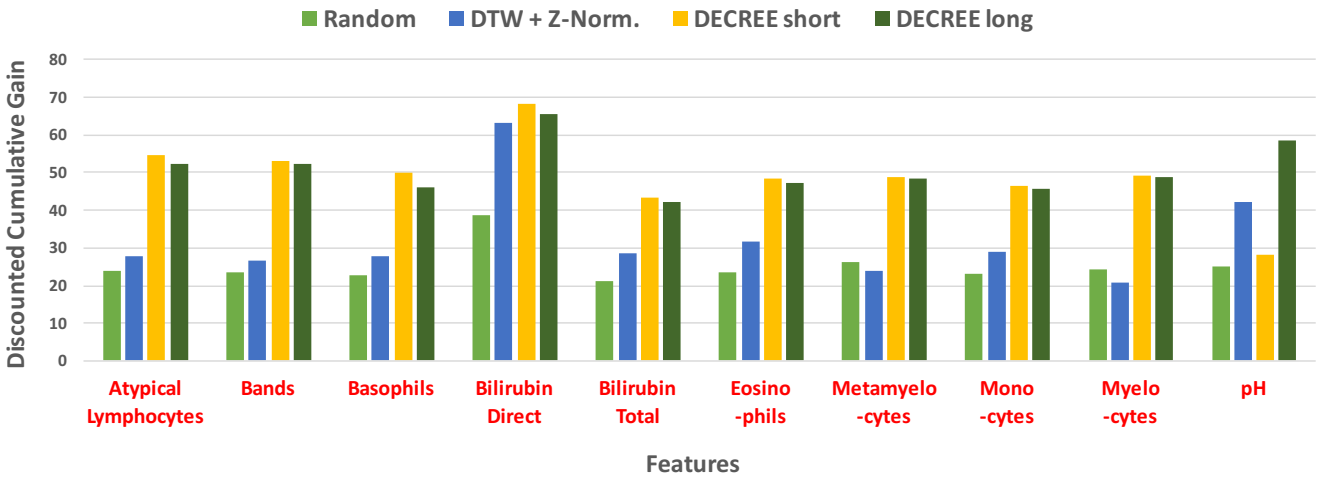


Figure 4: Discounted Cumulative Gain of DECREE, DTW and RANDOM for top-1000 pairs. 'DECREE short' refers to parameters : $\delta = 0.25$, $\rho = 0.5$ day and $\omega = 1.5$ day. 'DECREE long' refers to parameters : $\delta = 0.25$, $\rho = 1$ days and $\omega = 4$ days.

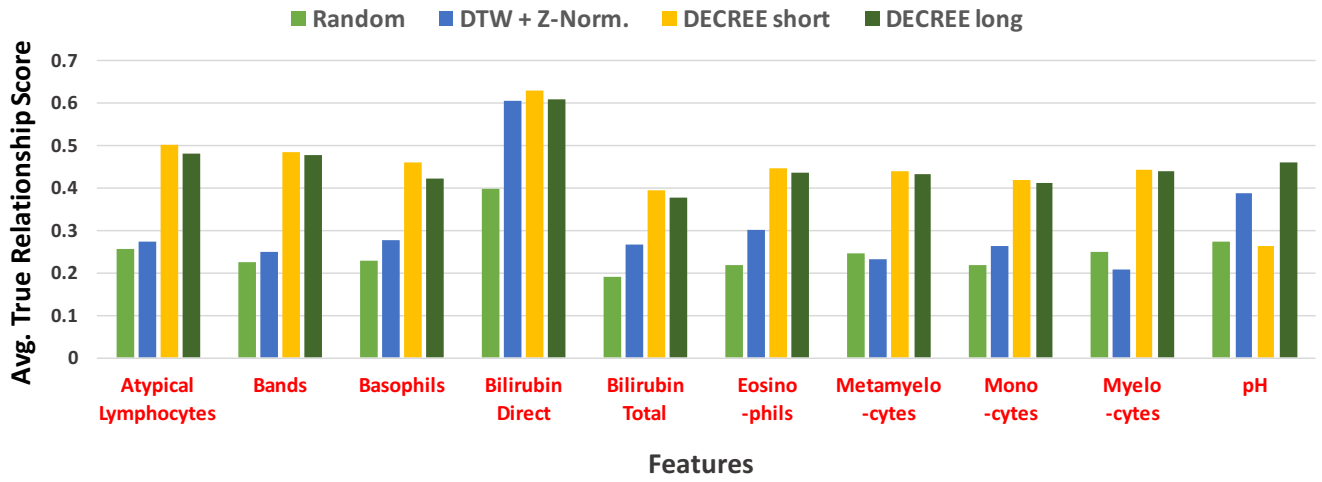


Figure 5: Average True Relationship Score (*ATRS*) of DECREE, DTW and RANDOM for top-1000 pairs. 'DECREE short' refers to the parameters : *local similarity distance threshold* $\delta = 0.25$, *local subsequence time window* $\rho = 0.5$ day and *search constraint time window* $\omega = 1.5$ day. 'DECREE long' refers to parameters : $\delta = 0.25$, $\rho = 1$ days and $\omega = 4$ days.

is consider as a normal value for Metamyelocytes. As a result, the patients having a time series containing multiple 0 values would have a smaller DTW distance. Same reasoning is applicable for Myelocytes. Note that, there are features like Temperature in which DTW and DECREE have *ATRS* close to RANDOM. This implies these features are not discriminative enough to identify relationships between diagnosis terms.

Reason for Low Accuracy of 'DECREE short' in pH : The *ATRS* of 'DECREE short' for pH is less than RANDOM and DTW. The reason is the subsequence window length s of half a day is too small for discovering *local similarities* between two time series T^e_{pH} for pH. If we keep the d distance threshold same and increase s and w, we observe with long size parameters of DECREE the *ATRS* of 'DECREE long' is better than RANDOM and DTW for pH. The accuracy of DECREE is dependent on the choice of parameters which is again dependent on the domain. We examine the sensitivity of parameters in subsection 5.5. From the Table 2, we know that the time-series of pH T^e_{pH} is relatively dense as compared to other T^e_m. In pH, we observe that the discriminative local similarity patterns cannot be captured when we use a small *local subsequence time window*(ρ) size of 0.5 day. But, as one can see from Fig. 9, when we keep this domain dependent ρ size between 1 day to 4 days, we see a sharp increase in accuracy.

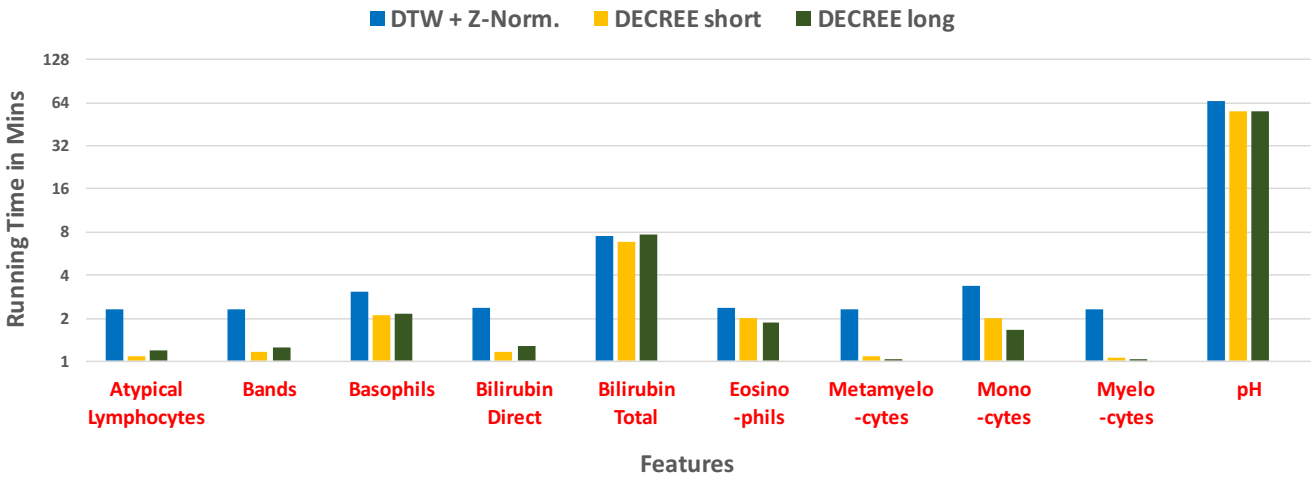


Figure 6: Running time of Decree-short, Decree-long and Dtw. Both Decree and Dtw are executed on same Spark cluster.

5.3 Running time

Figure 6 shows the comparison of running time of Decree and Dtw for discovering top-1000 pairs of related diagnosis terms. Both Decree and Dtw are executed on a same Spark cluster. Dtw along with z-normalization takes highest time whereas the Decree can produce good quality and better ranked results in less time. Random technique running time is not included because the pairs of entities are chosen randomly and no other processing is involved.

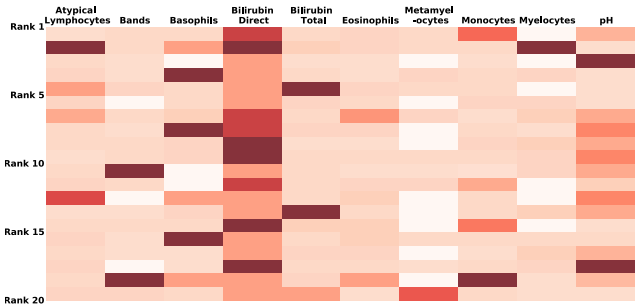
5.4 Case Study:

We present a case study related to Eosinophils laboratory test. An Eosinophil is a type of White Blood Cells (*WBC*) and Eosinophils count laboratory test is a blood test that measures the number of such *WBC*. The mined top-k list by Decree for feature Eosinophils show a pair of diagnosis terms "Acute myeloid leukemia, without mention of having achieved remission" and "Complications of transplanted bone marrow" at rank 5. The *True Relationship Score* between these two diagnosis terms is 0.34 which is low. However, the term myeloid refers to "relating to bone marrow". So both the diagnosis terms discovered by Decree are related to each other, even though ICD ontology organized both of those terms far from each other in hierachy. We showed an example where Decree can aid in the task of ontology enrichment.

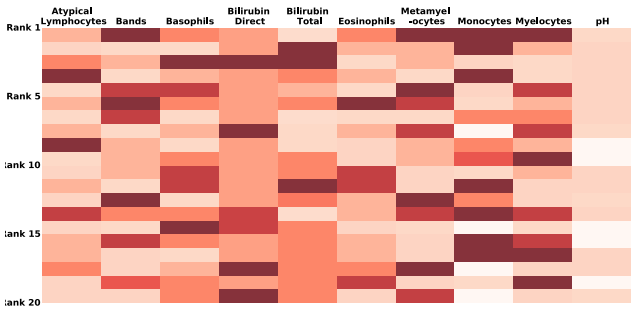
5.5 Parameter Sensitivity Experiments

The parameter sensitivity experiments for the features are shown in Figure 9. In first column, we keep the *local similarity distance threshold* δ constant at $\delta = 0.25$ while in second column $\delta = 5$. The x-axis is *local subsequence time window* ρ in days and y-axis is *search constraint time window* ω in days. Half a day is represented by 0.5. A bottom left cell in a parameter sensitivity matrix in the first column represents $\delta = 0.25$, $\rho = 0.5$ days, $\omega = 8$ days.

The value of a cell in the matrix represents the *Average True Relationship Score* for top-1000 related entities discovered by DECREE using a specific feature on right. The color scheme used in the experiments is shown at the bottom of the figure where 0 represents black color and 1 represents white color. Note that, $\omega \geq \rho$ hence the upper triangular values in parameter sensitivity matrix are equal to 0. We see that a large deviation from any parameter δ, ρ



(a) DTW



(b) Decree short

Figure 7: Heatmap of Top-20 diagnosis relationships of Dtw and 'Decree short'. Red color represents *True Relationship Score (TRS)* of 1 and White represents *TRS* of 0. The columns from left to right have same ordering as that of rows in Table 2 from top to bottom.

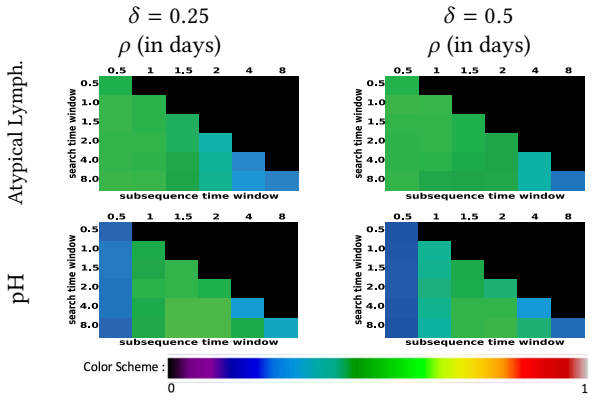


Figure 8: The parameter sensitivity experiments for the Atypical Lymphocytes and pH features

or ω results in significant deviation in the accuracy. For instance, a change of $\delta = 0.25$ to $\delta = 5$ while using feature "Bilirubin Direct" results in overall decrease in accuracy values while a change of $\delta = 0.25$ to $\delta = 0.5$ or $\delta = 1$ (Refer Figure 9) does not drastically change the spectrum of accuracy values. Similar trend is observed in other features with respect to δ. If one keeps ω and δ constant, the general trend seems to be: as the value of *local subsequence time window* ρ increases the accuracy value decreases but as one can see from the gradient this decrease in accuracy is not abrupt.

6 RELATED WORK

To the best of our knowledge, we are the first to propose the task of relating diagnosis terms by using time series as background data. Lara et. al. [8] define events with the help of *event definition language* which is developed using basic concepts of set theory, logic, algebra and descriptive statistics. Then, the authors with the help of mined events, classify electroencephalographic time series. The concept of event in [8] is different to that of *Local Similarity*. Batal et. al. [1] mines frequent recent temporal patterns from medical time-series and use these mined patterns for classfying the disorder of a patient into one of the eight categories of diabetes. We focus on a different task while analysing the medical time-series. Das Sarma et al. [4] creates a time-series for each entity based on the entity occurence in documents and detect co-bursting relationships between the time-series of entities. DECREE can not only detect co-bursting relationships but can also detect other complex relationships occuring in time-series as defined by *Local Similarity* (Def. 1). Bettini et. al. [3] present a method to discover temporal patterns in time sequences. The authors introduce event structure that is a user specified skeleton and it consists of a number of variables representing events and temporal constraints among these variables. Ye et. al. [13] propose time series shapelets, which are representative time-series subsequences. The representative quality of shapelets is useful for classification purpose but using shapelets to identify *relationship strength* between two time-series would be erroneous. A subsequence $T[s,t)$ might be common accross multiple class and hence would not be representative but presence of $T[s,t)$ among two time-series is an indictor of relationship between those two time-series.

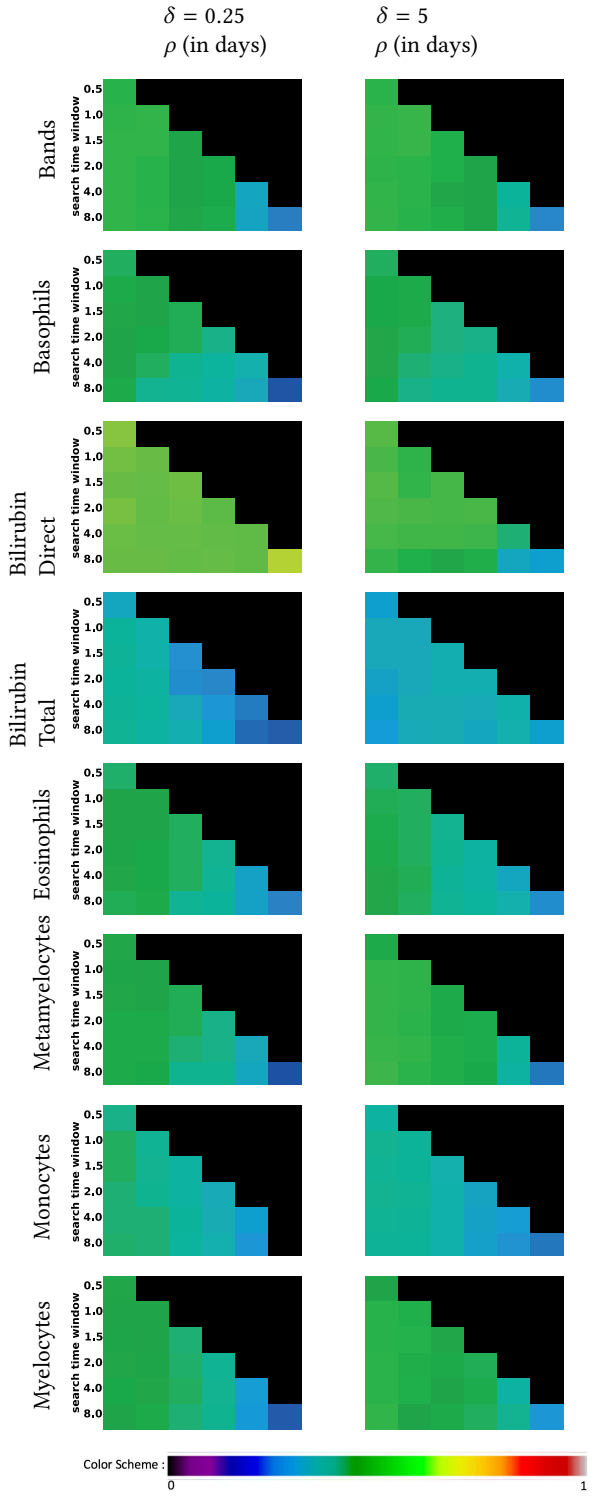


Figure 9: The parameter sensitivity experiments for the features mentioned in Table 2

7 CONCLUSION AND FUTURE WORK

In this paper, we demonstrated a data-centric approach that identifies the relationships between diagnostic terms using time-series analysis. We also proposed a time series analysis technique Decree that identifies related entities with better accuracy, ranking, and speed than Dynamic Time Warping. In the future, we plan to utilize multiple features for discovering related entities with the goal of improving accuracy.

REFERENCES

[1] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In *KDD*, 2012.

[2] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAIWS*, 1994.

[3] C. Bettini, X. Sean Wang, S. Jajodia, and J.-L. Lin. Discovering frequent event patterns with multiple granularities in time sequences. In *ITKDE*, 1998.

[4] A. Das Sarma, A. Jain, and C. Yu. Dynamic relationship and event discovery. In *WSDM*, 2011.

[5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. In *VLDB*, 2008.

[6] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. In *TOIS*, 2002.

[7] A. E. Johnson et al. Mimic-iii, a freely accessible critical care database. In *Scientific data*, 2016.

[8] J. A. Lara, D. Lizcano, A. Pérez, and J. P. Valente. A general framework for time series data mining based on event analysis. In *JBI*, 2014.

[9] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. In *JoDS*, 2005.

[10] C. Wang, L. Cao, and B. Zhou. Medical synonym extraction with concept space models. In *IJCAI*, 2015.

[11] F. Wang and J. Sun. Survey on distance metric learning and dimensionality reduction in data mining. In *Data Mining and Knowledge Discovery*, 2015.

[12] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, 1994.

[13] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *KDD*, 2009.

[14] A. H. Yee and A. A. Rabinstein. Neurologic presentations of acid-base imbalance, electrolyte abnormalities, and endocrine emergencies. In *Neurol Clin*, 2010.