Sequential Data Mining and its Applications to Pharmacovigilance

by

Xiao Qin

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Computer Science

April 23, 2019

APPROVED:

Professor Elke A. Rundensteiner
Worcester Polytechnic Institute
Advisor

Professor Xiangnan Kong
Worcester Polytechnic Institute
Committee Member

Professor Mohamed Y. Eltabakh
Worcester Polytechnic Institute
Committee Member

Professor Fei Wang
Weill Cornell Medicine
External Committee Member

Professor Craig Wills
Worcester Polytechnic Institute
Head of Department

Contents

Abstract

With the phenomenal growth of digital devices coupled with their ever-increasing capabilities of data generation and storage, sequential data is becoming more and more ubiquitous in a wide spectrum of application scenarios. There are various embodiments of sequential data such as temporal database, time series and text (word sequence) where the first one is *synchronous* over time and the latter two often generated in an *asynchronous* fashion. In order to derive precious insights, it is critical to learn and understand the behavior dynamics as well as the causality relationships across sequences.

Pharmacovigilance is defined as the science and activities relating to the *detection*, *assessment*, *understanding* and *prevention* of adverse drug reactions (ADR) or other drug-related problems. In the post-marketing phase, the effectiveness and the safety of drugs is monitored by regulatory agencies known as post-marketing surveillance. Spontaneous Reporting System (SRS), e.g., U.S. Food and Drug Administration Adverse Event Reporting System (FAERS), collects drug safety complaints over time providing the key evidence to support regularity actions towards the reported products. With the rapid growth of the reporting volume and velocity, data mining techniques promise to be effective to facilitating drug safety reviewers performing supervision tasks in a timely fashion.

My dissertation studies the problem of exploring, analyzing and modeling various types of sequential data within a typical SRS:

Temporal Correlations Discovery and Exploration. SRS can be seen as a temporal database where each transaction encodes the co-occurrence of some reported drugs and observed ADRs in a time frame. Temporal association rule learning (TARL) has been proven to be a prime candidate to derive associations among the objects from such temporal database. However, TARL is *parameterized* and computational expensive making it difficult to use for discovering interesting association among drugs and ADRs in a timely fashion. Worse yet, existing *interestingness* measures fail to capture the significance of certain types of association in the context of *pharmacovigilance*, e.g. drug-drug interaction (DDI) related ADR. To discover DDI related ADR using TARL, we propose an interestingness measure that aligns with the DDI semantics. We propose an interactive temporal association analytics framework that supports real-time temporal association derivation and exploration.

Anomaly Detection in Time Series. Abnormal reports may reveal meaningful ADR case which is overlooked by frequency-based data mining approach such as *association rule learning* where patterns are derived from frequently occurred events. In addition, the sense of abnormal or rareness may vary in different contexts. For example, an ADR, normally occurs to adult population, may rarely happen to youth population but with life threatening outcomes. *Local outlier factor* (LOF) is identified as a suitable approach to capture such local abnormal phenomenon. However, existing LOF algorithms and its variations fail to cope with high velocity data streams due to its high algorithmic complexity. We propose new local outlier semantics that leverage kernel

density estimation (KDE) to effectively detect local outliers from streaming data. A strategy to continuously detect top-N KDE-based local outliers over streams is also designed, called **KELOS** – the first linear time complexity streaming local outlier detection approach.

Text Modeling. Language modeling (LM) is a fundamental problem in many natural language processing (NLP) tasks. LM is the development of probabilistic models that are able to predict the next word in the sequence given the words that precede it. Recently, LM is advanced by the success of the recurrent neural networks (RNNs) which overcome the Markov assumption made in the traditional statistical language models. In theory, RNNs such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) can "remember" arbitrarily long span of history if provided with enough capacity. However, they do not perform well on very long sequences in practice as the gradient computation for RNNs becomes increasingly ill-behaved as the expected dependency becomes longer. One way of tackling this problem is to feed succinct information that encodes the semantic structure of the entire document such as latent topics as context to guide the modeling process.

Clinical narratives that describe complex medical events are often accompanied by meta-information such as a patient's demographics, diagnoses and medications. This structured information implicitly relates to the logical and semantic structure of the entire narrative, and thus affects vocabulary choices for the narrative composition. To leverage this meta-information, we propose a supervised topic compositional neural language model, called **MeTRNN**, that integrates the strength of supervised topic modeling in capturing global semantics with the capacity of contextual recurrent neural networks (RNN) in modeling local word dependencies.

Acknowledgments

I would like to express my sincere appreciation to my advisor Prof. Elke A. Rundensteiner and co-advisor Prof. Matthew O. Ward. I have been extremely lucky to have Prof. Rundensteiner as my advisor who cared so much about my work and life. I would like to thank for her patience, motivation, enthusiasm, and immense knowledge guiding me through my Ph.D. training process. I would like to express my deepest respect to Prof. Ward (1956 - 2014) who showed me the true meaning of enthusiasm for scientific research and more importantly led by example, being unbelievable brave and incredible positive towards his life.

My special thank you goes to my dissertation committee members Prof. Mohamed Y. Eltabakh, Prof. Xiangnan Kong and Prof. Fei Wang. I thank Prof. Eltabakh for providing valuable feedbacks on my early research work which eventually became my first full paper published in a reputable venue. I thank Prof. Kong who went out of his way to guide me through my learning process on machine learning and deep learning which is a core part of my dissertation. I am thankful to Prof. Wang, a distinguished scholar specializing in data analytics and its applications in health informatics for joining my dissertation committee and providing feedbacks on my dissertation.

I would like to thank all other research collaborators and co-authors of mine: Xika Lin, Ramoza Ahsan, Chris Botaish (WPI Undergraduate), Jason Whitehouse (WPI Undergraduate), Dr. Zhongqiang Chen (Yahoo!), Dr. Yuan Zhang (Yahoo!), Shenhong Zhu (Yahoo!), Dr. Lei Cao, Tabassum Kakar, Susmith Wunnava, Vimig Socrates (Case Western Reserve University), Amber Wallace (Lehigh University), Suranjan De (FDA), Sanjay Sahoo (FDA), Dr. Thang La (FDA), Brian McCarthy (WPI Undergraduate), Andrew Schade (WPI Undergraduate), Huy Quoc Tran (WPI Undergraduate), Brian Zylich (WPI Undergraduate), Cory Tapply (WPI Undergraduate), Derek Murphy (WPI Undergraduate), Daniel Yun (WPI Undergraduate), Oliver Spring (WPI Undergraduate), Prof. Lane Harrison, Prof. Samuel Madden (MIT), Chong Zhou, Dr. Cao Xiao (IQVIA) and Dr. Tengfei Ma (IBM). I also would like to thank all previous and current DSRG and XMDV members.

I am thankful to National Science Foundation (NSF), WPI and Oak Ridge Institute for Science and Education (ORISE) for funding my Ph.D. study at WPI.

Last but not least, I would like to thank my father Dr. Subin Qin (1956 - 2017), my mother Lingfen Kong and my wife Ruoyuan Gao for making everything possible and meaningful.

List of Publications during My Ph.D. Studies at WPI (in chronological order)

1. **Xiao Qin**, Cao Xiao, Tengfei Ma, Tabassum Kakar, Susmitha Wunnava, Xiangnan Kong, Elke Rndensteiner and Fei Wang. *Integrating Neural Language Model with Supervised Topic Modeling: Application to Clinical Narrative Modeling*. (In submission.)

2. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Xiangnan Kong, Elke Rundensteiner, Sanjay Sahoo and Suranjan De *Multi-Layered Learning for Information Extraction from Adverse Drug Event Narratives*. (In submission)

3. Tabassum Kakar, **Xiao Qin**, Cory M. Tapply, Oliver Spring, Derek Murphy, Daniel Yun, Elke A. Rundensteiner, Lane Harrison, Thang La, Sanjay K. Sahoo and Suranjan De. *ConText: In Pursuit of Evidence*. (In submission.)

4. Tabassum Kakar, **Xiao Qin**, Elke Rundensteiner, Lane Harrison, Sanjay Sahoo and Suranjan De. *DIVA: Towards Validation of Hypothesized Drug-Drug Interactions via Visual Analysis*. **EuroVis**'18.

5. **Xiao Qin**, Lei Cao, Elke A. Rundensteiner and Samuel Madden. *Scalable Kernel Density Estimation-based Local Outlier Detection over Large Data Streams*. **EDBT**'19.

6. Tabassum Kakar, **Xiao Qin**, Cory Tapply, Derek Murphy, Daniel Yun, Oliver Spring, Elke A. Rundensteiner, Lane Harrison, Thang La, Sanjay K. Sahoo and Suranjan De. *MedViz: Visual Analytics for Medication Error Detection*. **IVAPP**'19.

7. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Cansu Sen, Elke A. Rundensteiner and Xiangnan Kong. *Adverse Drug Event Detection from Electronic Health Records Using Hierarchical Recurrent Neural Networks with Dual-Level Embeddings*. **Drug Safety**'19, 1-10.

8. Brian Zylich, Brian McCarthy, Andrew Schade, Huy Quoc Tran, **Xiao Qin**, Tabassum Kakar and Elke A. Rundensteiner. *Drug-Drug Interaction Signal Detection from Drug Safety Reports*. **URTC**'18.

9. Tabassum Kakar, **Xiao Qin**, Susmitha Wunnava, Brian McCarthy, Andrew Schade, Huy Quoc Tran, Brian Zylich, Elke A. Rundensteiner, Lane Harrison, Sanjay K. Sahoo and Suranjan De. *DEVES: Interactive Signal Analytics for Drug Safety*. **CIKM**'18, 1891-1894.

10. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Elke A. Rundensteiner and Xiangnan Kong. *Deep Learning Strategies for Automatic Detection of Medication and Adverse Drug Events from Electronic Health Records.* **AMIA**'18.

11. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Elke A. Rundensteiner and Xiangnan Kong. *Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records*. **MADE**'18, 48-56.

12. **Xiao Qin**, Tabassum Kakar, Susmitha Wunnava, Brian McCarthy, Andrew Schade, Huy Quoc Tran, Brian Zylich, Elke A. Rundensteiner, Lane Harrison, Sanjay K. Sahoo and Suranjan De. *MeDIAR: Multi-Drug Adverse Reactions Analytics*. **ICDE**'18, 1565-1568.

13. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Xiangnan Kong, Elke Rundensteiner, Sanjay Sahoo and Suranjan De. *One Size Does Not Fit All: An Ensemble Approach Towards Information Extraction from Adverse Drug Event Narratives*. **HEALTHINF**'18, 176-188.

14. **Xiao Qin**, Tabassum Kakar, Susmitha Wunnava, Elke Rundensteiner and Lei Cao. *MARAS: Signaling Multi-Drug Adverse Reactions*. **KDD**'17, 1615-1623.

15. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Vimig Socrates, Amber Wallace and Elke Rundensteiner. *Towards Transforming FDA Adverse Event Narratives into Actionable Structured Data for Improved Pharmacovigilance*. **SAC**'17, 777-782.

16. **Xiao Qin**, Ramoza Ahsan, Xika Lin, Elke Rundensteiner, and Matthew Ward. *Interactive Temporal Association Analytics*. **EDBT**'16, 197-208.

17. **Xiao Qin**, Zhongqiang Chen, Yuan Zhang and Shenhong Zhu. *Death Hoax Detection in Query Suggestions*. **Yahoo! Tech Pulse**'15.

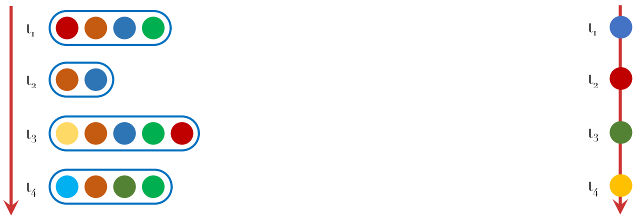
18. **Xiao Qin**, Ramoza Ahsan, Xika Lin, Elke Rundensteiner, and Matthew Ward. *iPARAS: Incremental Construction of Parameter Space for Online Association Mining*. **BigMine** workshop at **KDD**'14, JMLR 36 :149-165.

In particular, 1, 5, 12, 14, 16 and 18 are discussed in this dissertation. The other manuscripts are results of my collaborations with other scholars during my study at WPI.

1 Introduction

1.1 Sequential Data Mining

Data mining is a systematic process of extracting information from database and transforming it into an understandable structure for further usage. Over the past three decades, plentiful techniques have been invented and developed to discover useful patterns such as *frequent pattern mining*, *clustering*, *classification*, *outlier detection*, *regression*, *summarization* and etc. Beyond the core data analysis step, studies within the field of data mining also involve data processing and management aspects, data modeling and inference approaches, interestingness and evaluation metrics designs, computational complexity and efficiency considerations, data visualizations and etc.



(a) *Synchronous* sequence.

(b) *Asynchronous* sequence.

Figure 1: Two sequential data types.

The design of the data mining technique is primarily driven by the types of input data and the desirable transformation (output). My dissertation focuses on sequential data types, e.g. time-series data types and other ordered types such as text. These sequential data types can be categorized into *synchronous* and *asynchronous* sequence.

As illustrated in Figure 1(a), the *synchronous* type describes a sequence of events where an event consists of multiple objects occurring at the same time. An example of this type is retail database that stores customers' purchases. Each transaction records the products being checked out and is associated with a timestamp. A typical desired analysis on such data is *temporal association analysis* where the associations of different products over the time are discovered to support business decision making. On the other hand, the *asynchronous* type depicted in Figure 1(b) describes a sequence of objects occur in order without having the same timestamp. Examples of such type are *time series* that records the values of a variable over time, a piece of *text* where words are written in order according to a language and a *gene* which is a sequence of DNA or RNA which codes for a molecule. Many data mining tasks have been proposed and studied on this data type such as *trend analysis*, *language modeling* and *sequence generation*.

1.2 Applications in Pharmacovigilance

Pharmacovigilance is defined as the science and activities relating to the *detection*, *assessment*, *understanding* and *prevention* of adverse drug reactions (ADRs) or other drug-related problems. An adverse reaction corresponds to an unwanted and dangerous effect caused by the administration of a drug. According to the U.S. Food and Drug Administration (FDA) every year hundreds of thousands of people die because of these adverse reactions while over two million serious adverse reactions are reported every year. In the post-marketing phase the effectiveness and the safety of drugs is monitored by regulatory agencies known as post-marketing surveillance.

For early detection of novel ADRs which are not captured during the clinical trials, Spontaneous Reporting Systems (SRS) are designed to collect information on adverse events related to drugs reported by patients, health care professionals and drug manufacturers filed via mail, telephone and Internet. FDA Adverse Event Reporting System (FAERS) [] is one such system. Data collected from the surveillance programs is a useful resource to tap into ADRs. As thousands of new reports are added on a daily basis, discovering ADRS by aimlessly screening and analyzing all these reports is extremely difficult if not impossible. Therefore, computational methods, especially data mining techniques promise to be critical for identifying the most emerging ADR signals from massive reports. These signals which can be seen as ADR hypothesis along with the reports that derive these signals are then recommended to the drug safety evaluator for further investigation and validation.

A typical ADR report consists of structured fields and comment sections that are written in natural language. Structured fields record the most essential information about an incident such as the report date, patient's demographics, drugs and ADRs. The primary goal of SRS is to provide information to find emerging *unlabeled* ADR. One of the example tasks that can leverage such structured, temporal data to achieve the goal is temporal drug-ADR association analysis. That is, if a drug is highly associated with an ADR that is not supposed to be triggered by the usage of this drug according to the SRS database, then this drug-ADR pair may be a suspicious case that requires attention and further investigation. Another task that can use these structured information is performing report cluster/outlier detection where similar reports are gathered or special reports are identified to support systematic incidents exploration and monitoring.

Although the original report has structured fields, the unstructured narratives used for reporting an adverse event often contain information that is left blank in the structured fields. More importantly, these narratives are rich in detailed information regarding the adverse event. Automatically extracting information from the unstructured ADR report narratives into structured format is critical for advanced analytics and vital for timely detection, assessment and prevention of future incidents of ADRs. Many sequence modeling tasks have been studied to identify valuable information from the text such as *language modeling*, *sequence labeling* (named entity recognition) and *text generation*.

1.3 State-of-the-Art

Temporal Association Analytics. Temporal association rule learning (TARL) [111] is a technique that discovers temporal casual relationships among the items based upon their co-occurrence within a timeframe. It has been studied and extended to solve various problems including sequential association mining [42], cyclic association mining [80], stock trading rule mining [62], patent mining [101], clinical mining [106], image time series mining [48], software adoption and penetration mining [83], temporalutility mining [119], fuzzy temporal mining [61], and calendar association mining [111].

Lag in responsiveness is known to risk losing an analyst's attention during the exploration process. In applications like pharmacovigilance, such delay in decision making may prove to be the cause of public health crisis. Unfortunately, *temporal association mining algorithms* [7, 81] are known to be computationally intensive. To overcome this challenge, [111] pregenerates the *intermediate itemsets* that are subsequently used to derive the temporal associations instead of extracting them from the huge raw data store. With this promising one-time preprocessing strategy, the response time has been shown to be greatly reduced. However, the process of the final rule derivation remains a query-time task.

Temporal association mining algorithms are parametrized not only by traditional measures like *support*, *confidence* but also by *time-variant* measures [67, 95, 97]. Parameter settings used for one batch of data may produce insignificant rules for a newly incoming data batch. Thus the data analysts often must perform numerous successive trial-and-error iterations to find an appropriate parameter configuration out of a seemingly infinite number of possible settings. Existing state-of-the-art models tend to correspond to a blackbox [7, 38, 65, 81, 111] - providing little to no feedback about which parameter settings best capture the analyst's interest. To tackle this, [66] incorporates an indexing technique to swiftly produce parameter recommendations. However it is restricted to static data and thus does not support *time variant operations* essential for temporal association mining.

Temporal Outlier Detection. Finding *outliers* in streaming data is a fundamental task in many online applications ranging from fraud detection, network intrusion monitoring to system fault analysis.

LEAP [23] and Macrobase [8] scale distance-based and statistical-based outlier detection methods respectively to data streams where they rely on either the number of neighbors in a certain distance range or the frequency of each data point to detect outliers. More specifically in these works, a data point is considered to be an outlier if its neighbor count (or frequency) is lower than a *global* cut-off threshold. However, applying such a *global* cut-off threshold uniformly to the whole dataset is not ineffective in handling skewed datasets [33]. For example, a point with a small number of neighbors is not necessarily an outlier if it is located in a relative sparse subspace of the dataset. On the other hand, a point with a relative large number of neighbors might instead be an outlier, if it is located in a dense subspace and other points have many more neighbors than it.

Several methods [87, 99] have been proposed in recent years that leverage the concept of local outlier [18] to detect outliers from data streams. Local outlier is based on the observation that real world datasets tend to be skewed, where different subspaces of the data exhibit different distribution properties. It is thus often more meaningful to decide on the outlier status of a point based on its difference from the points in its *local neighborhood* as opposed to using a global density [23] or frequency [8] cutoff threshold to detect outliers [33]. More specifically, a point x is considered a *local outlier* if the probability density (PD) at x is low *relative* to that at the points in x's local neighborhood.

Unfortunately, existing streaming local outlier solutions [87, 99] are not scalable to high volume data streams. The root cause is that they measure the probability density at each point x based on the point's distance to its k nearest neighbors (kNN). Unfortunately, kNN is very sensitive to data updates, meaning that the insertion or removal of even a small number of points can cause the kNN of many points in the dataset to be updated. Since the complexity of the kNN search [18] is quadratic in the number of the points, significant resources may be wasted on a large number of unnecessary kNN re-computations. Therefore, those approaches suffer from high response time when handling high-speed streams. For example, it took [87, 99] 10 minutes to processing just 100k tuples as shown in their experiments.

Language Model and Text Generation. Language model can be classified into two categories, namely, *count based* and *continuous-space based* modeling. Count based models [53] such as traditional statistical language model, make an n-th order *Markov assumption* and estimate n-gram probabilities via counting and subsequent smoothing. The problem with such assumption is that the new combinations of n words that were not seen in the training corpus are likely to occur, thus causing zero probability being assigned frequently. Various smoothing methods such as modified Kneser-Ney smoothing [57] and Jelinek-Mercer smoothing [25] have been proposed to solve the data sparsity problem. In recent years, continuous space based models such as feed-forward neural probabilistic language models [11] (NPLMs) and recurrent neural network language models [73] (RNNs) are proposed. These Neural Language Models (NLM) solve the problem of data sparsity of the $n\text{-}gram$ model, by representing words as vectors (word embeddings) and using them as inputs to a NLM.

Language model has been applied to many NLP tasks including document modeling, information extraction text generation etc. Existing language models typically model locally coherent language that is on topic; however, overall they can miss information that should have been introduced or introduce duplicated or superfluous content. These errors are particularly common in situations where there are multiple distinct sources of input or the length of the output text is sufficiently long. [54] leverages the *attention mechanism* to control the content that must be included when generating desired topic. [28] combines *recurrent neural network* and *topic model* to model and generate text with hidden topics. In addition, documents are composed by different author and may describe totally different events. [76] proposes a variation of topic model that incorporates document level features for modeling. However, it does not model the language therefore cannot be used for tasks such as sequence labeling and text generation.

1.4 Research Challenges

Temporal Association Analytics. Given a time-variant data set containing n unique items, the maximum number of possible associations are bounded by $3^n - 2^n + 1$ [66]. The significance of associations may vary over time, as newly incoming data may bring new items and associations. Being able to quickly extract these associations and their behavior w.r.t different time horizons to answer analysts' requests is the key to providing an interactive mining experience. However, it is almost impossible to pre-generate all such information. Thus the system must have an efficient preprocessing strategy that pregenerates a minimal yet sufficient amount of information as its critical knowledge store to support interactive temporal association exploration.

Typical input parameters, such as *minimum support* and *confidence*, can be configured using any real number restricted to a certain range. Similarly, the time specification can be composed of one or multiple time periods along the continuous timeline. Clearly, an infinite number of possible parameter settings exists. Maintaining the corresponding ruleset for each parameter setting individually thus is impractical. Therefore, an efficient mechanism is needed to map the pregenerated temporal associations to the space of parameter settings.

The prominence level of an association may vary significantly for some associations while remaining stable for others. Such time-variant properties of parameter values may reveal important evolving patterns of an association in the evolving dataset. Yet keeping each single historical parameter value for each association is inefficient, resulting in large storage and search space. Therefore, a compact archive structure is needed to efficiently maintain the parameter values of the associations across time while supporting fast system access to retrieve any desired information.

To effectively rank the produced rules and therefore help the drug-safety evaluator concentrate on the rules most likely to be real MDARs, measures that effectively reflect the significance of the association between a set of drugs and a set of ADRs have to be provided. However, the off-the-shelf common used association measures such as *support*, *confidence* and *lift* (RR) [10] focus only on a single association rule based on the number of its occurrences, while the correlations among different rules have to be considered when measuring the significance of a rule to be a MDAR. For examples, if two rules contain the same ADRs and overlaps on the medicines, their significance might be influenced by each other. Therefore, we are in need of a customized measure to quantify the significance of an association in terms of its signaled MDARs.

Local Outlier Detection Over Data Stream. Effectively leveraging KDE in the streaming context comes with challenges. First, the effectiveness of KDE depends on several factors. In particular, both the kernel function and the smoothing parameter (commonly referred to as *bandwidth*) [121] have to be carefully selected to achieve a high accuracy for density estimation. Further, to ensure the effectiveness of KDE in multimodal distributions prevalent in real world datasets, customized density estimators have to be established for different data subspaces. This raises the problem of how to select relevant kernel centers to enable the inference of these different estimators. Making correct decisions on all these factors is complex. Worst yet, the distribution characteristics of a data stream evolve. Therefore, these factors would have to be continuously tuned to fit the data.

Furthermore, similar to kNN search, the complexity of KDE is quadratic in the number of points [102]. While the computational costs can be reduced by running the density estimation on kernel centers sampled from the input dataset, sampling leads to a trade-off between accuracy and efficiency. Although a low sampling rate can dramatically reduce the computational complexity, one must be cautious because the estimated probability density at each point may be inaccurate due to an insufficient number of kernel centers. On the other hand, a higher sampling rate will certainly lead to a better estimation of the density. However, computational costs of KDE increase quadratically with more kernel centers. With a large number of kernel centers, KDE would be at risk of becoming too costly to satisfy the stringent response time requirements of streaming applications.

Topically-coherent Text Generation. Existing studies [57, 25, 11] on language model focuses on modeling local coherent text based on n-gram Markov assumption. Although they can bed used to generate readable text in proper grammar, they lack of mechanism to control the topical coverage of the content. In many scenario, e.g. adverse drug reaction report narrative, a valid document contains certain information such indication, drug information and adverse events. Generating topically-coherent text is challenging since the generated must keeps track of what has been generated and what needs to be generated.

Recent work has focused on adapting neural network architectures to improve coverage [116] with application to generating customer service responses, such as hotel information, where a single sentence is generated to describe a few key ideas. [54] leverages *attention mechanism* with a controlled vocabulary to check if certain words are generated as expected. However, these techniques require explicit encoding and design to solve respective domain problem.

Documents are often composed by different authors and describing different events. The language used in these documents therefore may vary. TopicRNN [28] proposes a general solution that combines the strength of neural language model and topic model. However, it does not leverage the document level information for more customized modeling.

1.5 List of Proposed Solutions

Topic 1: Temporal Association Analytics. We propose the first *interactive temporal association rule mining* analytics framework called **TARA** [89] that enables analysts to explore associations across time and pinpoint appropriate parameter settings in a systematic way. The **TARA** model organizes the temporal associations in the space of query parameters. It abstracts the temporal associations at the coarse granularity of *time-aware stable regions* across multiple time periods. The **TARA** model is supported by *evolving parameter space* index structure that indexes *time-aware stable regions* along with the associated domination graph. To cope with the fast data, we propose an incremental parameter space index construction strategy [88] that can speed up the computation by orders of magnitude.

In the context of *pharmacovigilance*, existing semantics of the rule fail to capture the significance of certain types of association, e.g. drug-drug interaction (DDI) related ADR. To discover DDI related ADR using TARL, we propose an interestingness measure called *contrast* [92, 91] that aligns with the DDI semantics. Our experimental evaluations show that the *contrast* score allows the TARL to achieve high accuracy with significant less amount of rules.

Topic II: Temporal Local Outlier Detection. We propose new local outlier semantics that leverage kernel density estimation (KDE) to effectively detect local outliers from streaming data [90]. A strategy to continuously detect top-N KDE-based local outliers over streams is also designed, called **KELOS** – the first linear time complexity streaming local outlier detection approach. **KELOS** solves the effectiveness versus efficiency trade-off of KDE in the stream context by introducing the notion of abstract kernel centers. This concept could be applied to a much broader class of density estimation related stream mining tasks beyond outlier detection. Our extensive experiments using public datasets with outlier labels demonstrate the effectiveness of **KELOS** in detecting outliers while achieving several orders of magnitude performance gain in computational costs against the alternative approaches.

Topic III: Text Modeling and Generation. Clinical narratives that describe complex medical events are often accompanied by meta-information such as a patient's demographics, diagnoses and medications. This structured information implicitly relates to the logical and semantic structure of the entire narrative, and thus affects vocabulary choices for the narrative composition. We propose a neural language model called **MeTRNN** which enhances RNN-based language models' capability of establishing long-range dependencies by leveraging arbitrary document meta-information through their *implicit* influence via supervised latent topics and through *explicit* influence via a feature layer that directly connects to the RNN cells.

MeTRNN defines and explicitly models the text generative process based on the observation of the composition of the clinical narrative in an Electronic Health Record (EHR). **MeTRNN** captures the latent topics in text by leveraging the associated meta-information, which serves as the global context of the text that leads to better language modeling performance. To cope with various structured information in the EHRs, we propose a flexible supervised topic model component that can take on arbitrary meta-information. We design a joint model that connects sTMs to cRNNs with an end-to-end autoencoding variational Bayes inference method using the conditional variational autoencoder framework. It is a "black box" method that can be easily adjusted or extended. We demonstrate the effectiveness of **MeTRNN** in word prediction using publicly available text datasets as well as real world EHRs. **MeTRNN** achieves improvement in perplexity from 5% to 40% against baselines. We also conduct a case study that demonstrates **MeTRNN**'s ability to learn useful global context for better language modeling performance and more relevant topics to the structured meta-information.

1.6 Road Map

The dissertation is organized as follows. Chapter 1 first provides the introduction of this dissertation. Chapter 2 (Topic I) proposes the techniques for supporting interactive temporal association analytics and an interestingness measure to detect drug-drug interaction related adverse drug reactions. The local temporal outlier detection method for data streams is discussed in Chapter 3 (Topic II). Chapter 4 (Topic III) discusses the proposed RNN based language model for clinical text modeling and generation.

2 Temporal Association Analytics

Manuscript

1. **Xiao Qin**, Tabassum Kakar, Susmitha Wunnava, Brian McCarthy, Andrew Schade, Huy Quoc Tran, Brian Zylich, Elke A. Rundensteiner, Lane Harrison, Sanjay K. Sahoo and Suranjan De. *MeDIAR: Multi-Drug Adverse Reactions Analytics*. **ICDE**'18, 1565-1568.

2. **Xiao Qin**, Tabassum Kakar, Susmitha Wunnava, Elke Rundensteiner and Lei Cao. *MARAS: Signaling Multi-Drug Adverse Reactions*. **KDD**'17, 1615-1623.

3. **Xiao Qin**, Ramoza Ahsan, Xika Lin, Elke Rundensteiner, and Matthew Ward. *Interactive Temporal Association Analytics*. **EDBT**'16, 197-208.

4. **Xiao Qin**, Ramoza Ahsan, Xika Lin, Elke Rundensteiner, and Matthew Ward. *iPARAS: Incremental Construction of Parameter Space for Online Association Mining*. **BigMine** workshop at **KDD**'14, JMLR 36 :149-165.

2.1 Introduction

2.1.1 Motivation

Nowadays batches of data are continuously transmitted from a rich variety of sources including websites, mobile devices and other data sources, henceforth referred to as *evolving datasets*. Discovering associations and their dynamics hidden in such large evolving datasets has been recognized as critical for domains ranging from market products analysis, stock trend monitoring, targeted advertising to weather forecasting.

For example, in the retail businesses, the arrival of new fashions or gadgets may boost unprecedented sales while seasonal products may gain or lose customers' interest. Some products are purchased together more frequently in the days leading to a large sports event or during a traditional holiday like Thanksgiving. Companies such as Amazon, eBay, Walmart and other retail businesses apply temporal association mining techniques to their transaction logs to identify popular product combination at specific times and their behavior over time. Such information is critical for deciding the times when products can be placed together on a web page or configured into attractive bundle-offers to be used for recommendations to encourage sales.

Interactive data mining models, crucial for discovering knowledge from data, enable analysts to actively engage in the analysis process. State-of-the-art temporal association mining systems [7, 65, 81, 111], once supplied with a specific parameter setting, tend to generate the ruleset for each request from scratch. This one-at-the-time request model suffers from severe limitations described below.

2.1.2 Limitations of State-of-the-Art

Lack of instantaneous responsiveness. Lag in responsiveness is known to risk losing an analyst's attention during the exploration process. In applications like targeted ad placement such delay in decision making may prove to be the cause of missed business opportunities and thus a potentially huge loss in profit. Unfortunately, *temporal association mining algorithms* [7, 81] are known to be computationally intensive. To overcome this challenge, [111] pregenerates the *intermediate itemsets* that are subsequently used to derive the temporal associations instead of extracting them from the huge raw data store. With this promising one-time preprocessing strategy, the response time has been shown to be greatly reduced. However, the process of the final rule derivation remains a query-time task. This results in the shortcoming that the response times for mining such requests are not sufficient to support truly interactive exploration as confirmed by our experiments.

Lack of parameter recommendations. Temporal association mining algorithms are parametrized not only by traditional measures like *support* and *confidence* but also by *time-variant* measures [67, 95, 97]. Parameter settings used for one batch of data may produce insignificant rules for a newly incoming data batch. Thus the data analysts often must perform numerous successive trial-and-error iterations to find an appropriate parameter configuration out of a seemingly infinite number of possible settings. Existing state-of-the-art models tend to correspond to a blackbox [7, 38, 65, 81, 111] - providing little to no feedback about which parameter settings best capture the analyst's interest. To tackle this, [66] incorporates an indexing technique to swiftly produce parameter recommendations. However it is restricted to static data and thus does not support *time variant operations* essential for temporal association mining.

Lack of evolving ruleset comparison. Analysis of the data in finer time granularity may reveal that associations exist only in certain time periods. Some may fluctuate as new data arrives while others may remain stable. Furthermore, two seemingly similar parameter settings can generate different results. Systems like [7, 38, 81, 111] independently generate the ruleset for each parameter settings. Worse yet, analysts then have to go through a tedious process to manually investigate the results generated by different parameter settings to extract their differences. This can be extremely tedious and impractical for large data sets.

Lack of insights into the evolving associations. Given a parameter configuration, a system often generates a huge number of rules. Analysts would benefit from being able to quickly identify the most interesting ones, such as the most stable rules [67] within the last week, the most significant rules that occur every weekend, or the rules concerning specific products. Offering such rich insights into time-variant rule behavior would provide the analysts with the opportunity to leverage their domain knowledge to drive the discovery process. Unfortunately, most existing parameter-driven exploration systems [66, 95, 111] do not support the analyst in the discovery of such useful *time-sensitive* insights.

2.1.3 Research Challenges

To develop an interactive temporal analytic system, the following research challenges must be tackled.

Processing time-variant evolving data. Given a time-variant data set containing n unique items, the maximum number of possible associations are bounded by $3^n - 2^n + 1$ [66]. The significance of associations may vary over time, as newly incoming data may bring new items and associations. Being able to quickly extract these associations and their behavior w.r.t different time horizons to answer analysts' requests is the key to providing an interactive mining experience. However, it is almost impossible to pre-generate all such information. Thus the system must have an efficient preprocessing strategy that pregenerates a minimal yet sufficient amount of information as its critical knowledge store to support interactive temporal association exploration.

Managing temporal associations for all parameters. Typical input parameters, such as *minimum support* and *confidence*, can be configured using any real number restricted to a certain range. Similarly, the time specification can be composed of one or multiple time periods along the continuous timeline. Clearly, an infinite number of possible parameter settings exists. Maintaining the corresponding ruleset for each parameter setting individually thus is impractical. Therefore, an efficient mechanism is needed to map the pregenerated temporal associations to the space of parameter settings.

Maintaining parameter values for different time periods. The prominence level of an association may vary significantly for some associations while remaining stable for others. Such time-variant properties of parameter values may reveal important evolving patterns of an association in the evolving dataset. Yet keeping each single historical parameter value for each association is inefficient, resulting in large storage and search space. Therefore, a compact archive structure is needed to efficiently maintain the parameter values of the associations across time while supporting fast system access to retrieve any desired information.

Supporting advanced temporal association exploration. Rule mining algorithms tend to generate too many rules - making it extremely hard for the analysts to quickly identify

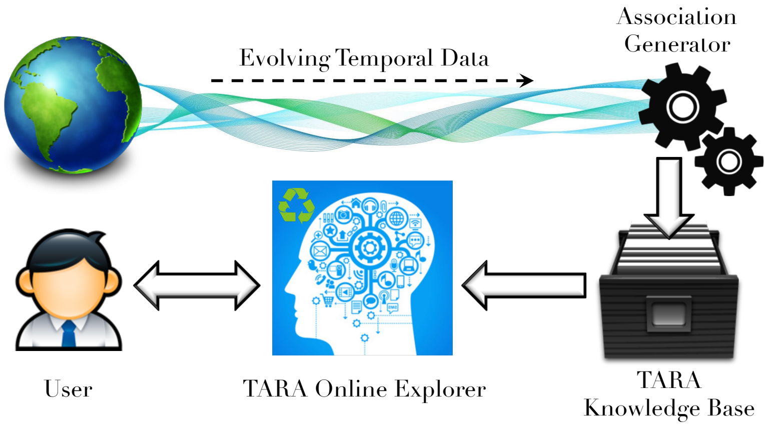


Figure 2: The TARA Approach.

the interesting ones. The problem of interestingness of temporal rules has been previously investigated [68, 97]. An interactive temporal association exploration system must integrate such interestingness measures to provide critical insights about the associations such as their evolving behaviors across time. The retrieved rules w.r.t particular parameter settings must be efficiently evaluated using these measures so that the instant responsiveness of the system is safeguarded.

2.1.4 The TARA Approach

We propose a novel temporal association rule analytics (**TARA**) framework that addresses the above challenges. The **TARA** infrastructure depicted in Figure 2 employs an offline preprocessing phase composed of Association Generator and Knowledge Base Constructor followed by TARA Online Explorer that enables analysts to interactively explore the evolving data with support by the knowledge base.

The Association Generator extracts temporal associations from the evolving data and compactly stores them in the Temporal Association Rule Archive (*TAR Archive*) of TARA knowledge base. Later, by request, the parameter values of a particular association w.r.t various fine granularities can be quickly computed without processing the raw data again. These pregenerated temporal associations are compressed into a knowledge-rich yet compact *evolving parameter space* (*EPS*) that encodes the relationships among the temporal associations. Next, the TARA knowledge base explicitly extracts and then models the distribution of the pregenerated temporal associations with respect to their parameters, e.g. support, confidence and time periods.

Beyond achieving speedup in response time, the online processing strategies leverage the *EPS* index to offer analysts an innovative "rule-centric panorama" into the temporal associations present within the evolving dataset. The framework supports rich classes of novel exploration operations from time-travel queries and parameter recommendations to evolving ruleset comparisons.

2.1.5 Contributions

Key contributions of this work include:

- We propose the first *interactive temporal association rule mining* analytics framework called **TARA** that enables analysts to explore associations across time and pinpoint appropriate parameter settings in a systematic way.
- The **TARA** model organizes the temporal associations in the space of query parameters. It abstracts the temporal associations at the coarse granularity of *time-aware stable regions* across multiple time periods.
- The **TARA** model is supported by *evolving parameter space* (*EPS*) index structure that indexes *time-aware stable regions* along with the associated domination graph. TARA offers efficient algorithms for offline *EPS* index construction.
- For the rules generated, we design a temporal association rule archive, called *TAR Archive*, that compactly encodes the parameter values of each rule across time. Our specially designed encoding and decoding strategies achieve fast access to the requested information from this archive.
- We propose a rich set of novel temporal rule exploration operations beyond traditional temporal rule mining. Effective strategies for the online processing of the proposed operations that leverage our precomputed TARA index structures are provided.
- TARA framework supports the exploration of the associations at coarser or finer time granularities by roll-up and drill down operations. We provide a theoretical bound on the approximation of the solution under roll-up operations.
- Our extensive experiments using IBM Quest [5], *retail* [19] and *webdocs* [70] datasets demonstrate that **TARA** is 3 to 5 orders of magnitude faster than its state-of-the-art competitors for traditional temporal association mining, while in addition supporting novel analytics within milliseconds.

2.2 Foundation

2.2.1 Temporal Association Rule

$\mathcal{T} = \{..., t_i, ..., t_j, ...\}$ denotes a set of **times**, countably infinite, over which a linear order ¡$_{\mathcal{T}}$ is defined, where t_i ¡$_{\mathcal{T}}$ t_j means t_i occurs strictly before t_j. Let $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ represent a set of **items**. $\mathcal{D} = \{d_1, d_2, ..., d_m\}$ is a collection of subsets of I called the **transaction**

database. Each **transaction** d_i in $\mathcal{D}$ is a set of items such that $d_i \subseteq \mathcal{I}$. Each d_i has an associated timestamp t_j, denoted by $d_i.time$ = t_j. Let $\mathcal{X} \subseteq \mathcal{I}$ be a set of items, called **itemset**. If $\mathcal{X} \subseteq d_i$, d_i *contains* $\mathcal{X}$. If the cardinality of $\mathcal{X}$ is k, $\mathcal{X}$ is called a **k-itemset**. Given a closed **time period** $[t_i, t_j]$ where t_i ¡$_{\mathcal{T}}$ t_j, then the set of transactions in the range $[t_i, t_j]$ of $\mathcal{D}$ that *contain* $\mathcal{X}$ is indicated by $\mathcal{F}(\mathcal{X}, \mathcal{D}, [t_i, t_j])$ = $\{d_k \mid d_k \in \mathcal{D} \wedge t_i \leq d_k.time \leq t_j \wedge \mathcal{X} \subseteq d_k\}$.

Definition 1. *A* ***temporal association rule*** *is an expression of the form* $\mathcal{R}^{[t_i,t_j]} \equiv (\mathcal{X} \Rightarrow \mathcal{Y})$, *where* $\mathcal{X} \subseteq \mathcal{I}$, $\mathcal{Y} \subseteq \mathcal{I} \setminus \mathcal{X}$, *and* $[t_i, t_j]$ *indicates that* $\mathcal{R}$ *is derived from all the transactions in* $\mathcal{D}$ *whose timestamps fall into* $[t_i, t_j]$.

A **temporal association rule** defaults to the traditional association rule if the *time period* is set to the entire timeline. This time restriction $[t_i, t_j]$ empowers the data analysts to discover associations that are not significant throughout the entire data set. Moreover, an association may reappear in multiple time periods expressing some periodicity. Furthermore, the association may behave differently in terms of its measured values. The evolution of the associations over time can lead to insightful observations [67].

2.2.2 Interestingness Measures

Many measurements [97] have been proposed to evaluate the interestingness of associations. Out of these measurements, we work with the most common measures of *support* and *confidence* to demonstrate the key principles of our framework, though others can be plugged in the future.

$$Support(\mathcal{R}^{[t_i,t_j]}) = \frac{|\mathcal{F}(\mathcal{X} \cup \mathcal{Y}, \mathcal{D}, [t_i, t_j])|}{|\mathcal{F}(\emptyset, \mathcal{D}, [t_i, t_j])|} \tag{1}$$

$$Confidence(\mathcal{R}^{[t_i,t_j]}) = \frac{|\mathcal{F}(\mathcal{X} \cup \mathcal{Y}, \mathcal{D}, [t_i, t_j])|}{|\mathcal{F}(\mathcal{X}, \mathcal{D}, [t_i, t_j])|} \tag{2}$$

$$Lift(\mathcal{R}^{[t_i,t_j]}) = \frac{|\mathcal{F}(\mathcal{X} \cup \mathcal{Y}, \mathcal{D}, [t_i, t_j])| \times |\mathcal{F}(\emptyset, \mathcal{D}, [t_i, t_j])|}{|\mathcal{F}(\mathcal{X}, \mathcal{D}, [t_i, t_j])| \times |\mathcal{F}(\mathcal{Y}, \mathcal{D}, [t_i, t_j])|} \tag{3}$$

The *support* (Formula 1) describes the proportion of the transactions that *contain* all items in the association. *confidence* (Formula 2) describes the probability of finding the *consequent* $\mathcal{Y}$ of the association under the condition that these transactions also *contain* the *antecedent* $\mathcal{X}$. It is a maximum likelihood estimate of the conditional probability $P(\mathcal{Y}|\mathcal{X})$. *Lift* (Formula 3) measures how many times more often $\mathcal{X}$ and $\mathcal{Y}$ occur together than expected if they are statistically independent.

2.3 Interestingness Measure for Finding Drug-Drug Interactions

Let $\mathcal{I}_{Drug} = \{d_1, d_2, ..., d_o\}$ and $\mathcal{I}_{ADR} = \{a_1, a_2, ..., a_u\}$ represent a set of drugs and a set of ADRs where $\mathcal{I}_{Drug} \cap \mathcal{I}_{ADR} \equiv \emptyset$. $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ is a collection of ADR reports. Each report $t_i \equiv \mathcal{D}_i \cup \mathcal{A}_i$ contains a drug set $\mathcal{D}_i$ where $\mathcal{D}_i \subseteq \mathcal{I}_{Drug}$ and an ADR set $\mathcal{A}_i$ where $\mathcal{A}_i \subseteq \mathcal{I}_{ADR}$. Since we are only interested in modeling the associations from a set of drugs to a set of ADRs in a collection of ADR reports, we define the Drug-ADR association as below.

Definition 2. *A **Drug-ADR association** is an expression of the form $\mathcal{R} \equiv \mathcal{D} \Rightarrow \mathcal{A}$ where $\mathcal{D} \subseteq \mathcal{I}_{Drug}$, $\mathcal{A} \subseteq \mathcal{I}_{ADR}$ and $\mathcal{I}_{Drug} \cap \mathcal{I}_{ADR} \equiv \emptyset$.*

Irrelevant Association. If the traditional association rule model were to be directly applied on the ADR reports $\mathcal{T}$, the ARL algorithm can possibly generate $3^{o+u} - 2^{o+u} + 1$ ($\mathcal{O}(3^n)$ where $n = o + u$) associations where o and u denote the total number of unique drugs and ADRs respectively. However, based on Definition 2, the number of possible Drug-ADR associations instead corresponds to:

$$|2^{\mathcal{I}_{Drug}} \times 2^{\mathcal{I}_{ADR}}| = \sum_{k=1}^{o} \binom{o}{k} \times \sum_{k=1}^{u} \binom{u}{k} = (2^o - 1) \times (2^u - 1). \tag{4}$$

According to Formula 4, the number of possible Drug-ADR associations ($\mathcal{O}(2^n)$ where $n = o + u$) is much smaller than $\mathcal{O}(3^n)$. The associations that do not confirm the defined Drug-ADR expression are *irrelevant*, therefore need to be pruned in the learning process. Also, since we study MDARs in this work, we focus on the Drug-ADR associations which contain at least two drugs in the antecedent.

2.3.1 Non-spurious Drug-ADR Association

Without pre-established dependency constraints among items, existing ARL algorithms [117] consider every possible combination of items that appears in a transaction as an *itemset*. This results in a huge amount of *redundant* [120, 9, 85] even *misleading* associations in the context of signaling ADRs from ADR reports as we show below.

2.3.2 Types of Drug-ADR Associations

Explicitly Supported Drug-ADR Association. Let us consider an ADR report $t_i \equiv \mathcal{D}_i \cup \mathcal{A}_i$ with a set of drugs $\mathcal{D}_i \equiv \{d_1, d_2, d_3\}$ and a set of ADRs $\mathcal{A}_i \equiv \{a_1, a_2\}$. This particular ADR report explicitly establishes the association between $\mathcal{D}_i$ and $\mathcal{A}_i$, expressed by the association $\mathcal{R}_1 \equiv (d_1 \wedge d_2 \wedge d_3) \Rightarrow (a_1 \wedge a_2)$. However, based upon this single report, traditional ARL would generate 24 variants of Drug-ADR associations ($(3^2 - 1) \times (2^2 - 1)$), such as $(d_1 \wedge d_2) \Rightarrow (a_1)$, $(d_1 \wedge d_3) \Rightarrow (a_2)$ etc. including $\mathcal{R}_1$. All of these associations, except $\mathcal{R}_1$, are **partial interpretations** of the report, randomly leaving out certain item(s), e.g., some drugs or some ADRs mentioned in the report. In many scenarios, these associations could be misleading unless there is additional evidence to support them. For example, $\mathcal{R}_2 \equiv d_1 \Rightarrow a_2$ tells us that taking d_1 might lead to a_2. This may however not be true in our context since this report does not *explicitly indicate* that drug d_1 by itself will lead to ADR a_2 therefore cannot be confirmed by this ADR report.

Definition 3. *A Drug-ADR association $\mathcal{R} \equiv \mathcal{D} \Rightarrow \mathcal{A}$ is **explicitly supported** by a collection of ADR reports $\mathcal{T}$ if there exists at least one report $t_i \in \mathcal{T}$ where $t_i \equiv \mathcal{D}_i \cup \mathcal{A}_i$ such that $t_i \equiv \mathcal{D} \cup \mathcal{A}$.*

If a Drug-ADR association is *explicitly* supported, according to definition 3, at least one report must exist that refers exactly to drugs and ADRs in the association and no additional ones. Other reports that contain these drugs and ADRs can be used as evidence to measure the significance of this association.

Implicitly Supported Drug-ADR Association. In addition to t_i in the last example, let us consider adding another ADR report $t_j \equiv \mathcal{D}_j \cup \mathcal{A}_j$ with a set of drugs $\mathcal{D}_j \equiv \{d_1, d_2, d_4\}$ and a set of ADRs $\mathcal{A}_i \equiv \{a_1, a_2\}$. According to Definition 3, $\mathcal{R}_3 \equiv (d_1 \wedge d_2 \wedge d_4) \Rightarrow (a_1 \wedge a_2)$ is *explicitly* supported by $\mathcal{T}$. Although the Drug-ADR association $\mathcal{R}_4 \equiv (d_1 \wedge d_2) \Rightarrow (a_1 \wedge a_2)$ is a **partial interpretation** of t_i or t_j, it may be of interest to the drug safety evaluator since it involves the **intersection** of two reports which can be interpreted as a commonly prescribed drug combination or a commonly caused ADRs. The Drug-ADR associations formed by the intersection of multiple reports such as $\mathcal{R}_4$ are defined as *implicitly supported* Drug-ADR associations:

Definition 4. *A Drug-ADR association $\mathcal{R} \equiv \mathcal{D} \Rightarrow \mathcal{A}$ is **implicitly supported** by a collection of ADR reports $\mathcal{T}$ if there exist at least two ADR reports $t_i, t_j \in \mathcal{T}$ where $i \neq j, t_i \not\equiv t_j, t_i \equiv \mathcal{D}_i \cup \mathcal{A}_i$ and $t_j \equiv \mathcal{D}_j \cup \mathcal{A}_j$ such that $t_i, t_j \not\equiv \mathcal{D} \cup \mathcal{A}$, $\mathcal{D} \equiv \mathcal{D}_i \cap \mathcal{D}_j$ and $\mathcal{A} \equiv \mathcal{A}_i \cap \mathcal{A}_j$.*

According to Definition 4, if a Drug-ADR association is *implicitly* supported, it models an association between a commonly prescribed drug combination and commonly caused ADRs suggested by at least two reports and it is not *explicitly* supported. If a Drug-ADR association is neither *explicitly* nor *implicitly* supported, it is a **spurious association** which must be treated with caution as it may convey misleading information. Next, we will discuss how our system identifies these associations.

2.3.3 Learning Non-spurious Drug-ADR Association

$\mathcal{S}_{exp}$ and $\mathcal{S}_{imp}$ denote complete sets of *explicitly* and *implicitly* supported Drug-ADR associations learned from a collection of ADR reports $\mathcal{T}$. Below we show that identifying $\mathcal{S}_{exp} \cup \mathcal{S}_{imp}$ is equivalent to identifying *closed* associations [85] from all possible Drug-ADR associations in $\mathcal{T}$. *Closed* associations [9] compactly represent the same information as the full set of all possible associations and can be used to recover the full set. The notion of a *closed* association is defined as below:

Definition 5. *An association $\mathcal{R}_i \equiv \mathcal{X}_i \Rightarrow \mathcal{Y}_i$ is called* **closed** *in a set of transactions $\mathcal{T}$ if there does not exist an association $\mathcal{R}_j \equiv \mathcal{X}_j \Rightarrow \mathcal{Y}_j$ where $i \neq j$ such that $\mathcal{X}_i \cup \mathcal{Y}_i \subset \mathcal{X}_j \cup \mathcal{Y}_j$ and $|\mathcal{X}_i \cup \mathcal{Y}_i|$ = $|\mathcal{X}_j \cup \mathcal{Y}_j|$.*

According to Definition 5, if an association $\mathcal{R}_i$ is not *closed* in a dataset, there exists another association $\mathcal{R}_j$ with additional items (richer information) which is also contained by the same set of transactions. For example, for associations $\mathcal{R}_1 \equiv (i_1 \wedge i_2) \Rightarrow (i_3 \wedge i_4)$ and $\mathcal{R}_2 \equiv (i_1) \Rightarrow (i_3 \wedge i_4)$ where i represents an item, if $|\{i_1, i_2, i_3, i_4\}|$ = $|\{i_1, i_3, i_4\}|$, this means that $\mathcal{R}_1$ and $\mathcal{R}_2$ are contained by the same set of transactions. Regardless whether or not $\mathcal{R}_1$ is *closed*, $\mathcal{R}_2$ is not closed since it only presents partial information of $\mathcal{R}_1$.

Let $\mathcal{S}_{Drug-ADR}$ denote a complete set of Drug-ADR associations learned from a collection of ADR reports $\mathcal{T}$ and $\mathcal{S}^*_{Drug-ADR}$ be the complete set of *closed* Drug-ADR associations in $\mathcal{S}_{Drug-ADR}$. We have the following claim.

Lemma 1. *The closed Drug-ADR association set $\mathcal{S}^*_{Drug-ADR} \equiv \mathcal{S}_{exp} \cup \mathcal{S}_{imp}$ where $\mathcal{S}^*_{Drug-ADR}$, $\mathcal{S}_{exp}$ and $\mathcal{S}_{imp}$ are learned from the same collection of ADR reports $\mathcal{T}$.*

Proof. The proof is bi-directional. First, if a Drug-ADR association is *closed*, it is either *explicitly* or *implicitly* supported. Second, if a Drug-ADR association is either *explicitly* or *implicitly* supported, it must be *closed*.

First, consider a Drug-ADR association $\mathcal{R} \equiv \mathcal{D} \Rightarrow \mathcal{A}$, if $\mathcal{R}$ is *closed* then there does not exist an $\mathcal{R}_i$ such that $\mathcal{R}_i$ has additional items beyond $\mathcal{R}$ and is contained by the same set of ADR reports as $\mathcal{R}$. There are two possibilities causing such non-existence: (1) no report exists that contains more items than $\mathcal{D} \cup \mathcal{A}$ which makes $\mathcal{R}$ *explicitly* supported; (2) $\mathcal{D} \cup \mathcal{A}$ is an intersection of multiple reports and all $\mathcal{R}_i$ with additional items are of course also contained in less reports; If there is a report among them that contains the exact same items in $\mathcal{R}$ then $\mathcal{R}$ is *explicitly* supported, otherwise it is *implicitly* supported.

Second, if $\mathcal{R}$ is *explicitly* supported, either (1) there exists no report with additional items in $\mathcal{R}$ which makes $\mathcal{R}$ *closed* because there is no $\mathcal{R}_i$ with additional items that can be learned from the reports; or (2) in addition to the report(s) that contain the exact items in $\mathcal{R}$, there are reports with more items; But this will make the R_i with additional items be contained by less amount of reports than $\mathcal{R}$; Therefore, $\mathcal{R}$ is *closed*. If $\mathcal{R}$ is *implicitly* supported, it contains the interaction of multiple reports, then all the R_i with additional items are contained by less reports; Therefore $\mathcal{R}$ is *closed*. □

We use Lemma 1 as theoretical foundation to efficiently identify non-spurious Drug-ADR associations.

2.3.4 Contextual Association Cluster

Table 1: Example of a Contextual Association Cluster of $\mathcal{R}$

$\mathcal{R}$	[Furosemide] [Isosorbide] [Aspirin] ⇒ [Myocardial Infarction]
$\tilde{\mathcal{R}}^2$	$\tilde{\mathcal{R}}^2_1 \equiv$ [Furosemide] [Isosorbide] ⇒ [Myocardial Infarction]
	$\tilde{\mathcal{R}}^2_2 \equiv$ [Furosemide] [Aspirin] ⇒ [Myocardial Infarction]
	$\tilde{\mathcal{R}}^2_3 \equiv$ [Isosorbide] [Aspirin] ⇒ [Myocardial Infarction]
$\tilde{\mathcal{R}}^1$	$\tilde{\mathcal{R}}^1_1 \equiv$ [Furosemide] ⇒ [Myocardial Infarction]
	$\tilde{\mathcal{R}}^1_2 \equiv$ [Isosorbide] ⇒ [Myocardial Infarction]
	$\tilde{\mathcal{R}}^1_3 \equiv$ [Aspirin] ⇒ [Myocardial Infarction]

Next, we introduce how MARAS measures non-spurious Drug-ADR associations that contain multiple drugs to signal MDARs. Existing measures [10] including *support*, *confidence* and *lift* (RR) evaluate the strength of the association between two set of items. However, they lack the ability to verify whether this strong association is already implied by a subset of the antecedent. Such a domination from a subset of the drug antecedents would weaken the MDAR signal. For example, if the ADRs are already highly associated with an individual drug in the given combination of drugs of the association, it means that the ADRs are likely caused by this particular drug or subset of drugs instead of the larger MDAR.

To measure this notion of *exclusiveness* of the association between drugs and ADRs, any association between a subset of drugs and the ADRs needs to be considered. These related associations are henceforth referred to as the **contextual** associations of the target association.

Definition 6. *A Drug-ADR association $\mathcal{R}_i \equiv \mathcal{D}_i \Rightarrow \mathcal{A}_i$ is a **contextual association** of a Drug-ADR Association $\mathcal{R}_j \equiv \mathcal{D}_j \Rightarrow \mathcal{A}_j$ if and only if $\mathcal{D}_j \subset \mathcal{D}_i$ and $\mathcal{A}_i \equiv \mathcal{A}_i$.*

Based on Definition 6, we define the **Contextual Association Cluster** (CAC) of a target Drug-ADR association.

Definition 7. *A* ***Contextual Association Cluster*** $\mathcal{C} \equiv \{\mathcal{R}, \tilde{\mathcal{R}}_1,...,\tilde{\mathcal{R}}_n\}$ *includes an explicitly or implicitly supported Drug-ADR association* $\mathcal{R} \equiv \mathcal{D} \Rightarrow \mathcal{A}$ *and its contextual associations such that* $\bigcup_{i=1}^{n} \tilde{\mathcal{D}}_i \equiv \mathbb{P}(\mathcal{D}) - \{\emptyset, \mathcal{D}\}$ *where* $\tilde{\mathcal{D}}_i$ *is antecedent of the contextual association* $\tilde{\mathcal{R}}_i$ *and* $\mathbb{P}(\mathcal{D})$ *is the power set of* $\mathcal{D}$*.* $\mathcal{R}$ *is called* ***target*** *association.*

Table 1 shows an example of the CAC of a target Drug-ADR association $\mathcal{R}$ which represents the MDAR signal. The CAC is organized based on the cardinality of the antecedent. The number n in $\tilde{\mathcal{R}}^n$ refers to the number of drugs in the association. In this example, $\mathcal{R}$ has 3 drugs. Hence, there are 6 contextual associations in CAC. MARAS uses CAC to evaluate the interestingness of the target Drug-ADR association that contains multiple drugs in terms of signaling the most severe MDARs.

2.3.5 Contrast Score for MDAR Signal

To measure if a Drug-ADR association encodes a strong signal that indicates a severe MDAR, two factors need be taken into consideration. First, how strong the association of ADRs is with the drug combination and second, how strong the association of ADRs is with the individual or subset of drugs. As explained in Section 2.3.4, if ADRs are caused by the interaction of a drug combination then not only the ADRs must be strongly associated with the drug combination but also any subset of these drugs should only be weakly associated with the particular ADRs.

For the first factor, MARAS adopts the *confidence* model that represents a maximum likelihood estimate of the conditional probability $P(\mathcal{A}|\mathcal{D})$ for a Drug-ADR association $\mathcal{R}$. It models the strength of the association between the antecedent and consequent. High *confidence* indicates strong association while low *confidence* indicates weak association. For the second factor, we first defined the CAC introduced in Section 2.3.4. A CAC includes a target association that represents the MDAR signal along with all its contextual associations that represent the associations between the target ADRs and the subsets of the target drugs. The MDAR signal is strongest if the target association has high *confidence* and all of its contextual associations in the cluster have low *confidence*. To quantify such a contrast that captures the intuition of the MDAR phenomenon, as discussed in Section 2.3.4, we propose the *contrast* measure.

Let $\mathcal{C} \equiv \{\mathcal{R}, ..., \tilde{\mathcal{R}}_j^i, ...\}$ represent a CAC, with $\mathcal{R}$ the target association and $\tilde{\mathcal{R}}_j^i$ its contextual associations where i denotes the number of drugs in the association and j is used to distinguish between different contextual associations with the same amount of drugs i. $\mathcal{P}_c(\mathcal{R})$ denotes the *confidence* of an association $\mathcal{R}$. The MDAR signal is strong if the *confidence* of $\mathcal{R}$ is significantly higher than any *confidence* of its contextual associations.

$$contrast_{max}(\mathcal{C}) = \mathcal{P}_c(\mathcal{R}) - max(\mathcal{P}_c(\tilde{\mathcal{R}}_j^i)). \tag{5}$$

A negative *contrast*$_{max}$ value means that a subset of drugs is more likely to cause the ADRs then the actual target set. This idea is similar to the *improvement* measure proposed by Bayardo *et al.* [52]. However, only considering the contextual association with the highest *confidence* deprives us of the opportunity to differentiate more complex cases. For example, even if two MDAR signals share the same *contrast* value, the one with more higher *confidence* contextual associations may be less interesting than the other one because more drugs may cause the same ADRs showing a weaker sign of the MDAR. To utilize the full context in the evaluation of the MDAR signal, an alternative solution would be to measure the difference between the *confidence* of the target association and the average *confidence* of its contextual associations:

$$contrast_{avg}(\mathcal{C}) = \mathcal{P}_c(\mathcal{R}) - \frac{1}{|\mathcal{C}| - 1} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathcal{P}_c(\tilde{\mathcal{R}}_j^i). \tag{6}$$

The shortcoming of this solution is that it falsely weakens the negative effect of any contextual association with a high *confidence*. For example, let us consider two CAC cases $\mathcal{C}_1 \equiv \{\mathcal{R}_, \tilde{\mathcal{R}_1^1}, \tilde{\mathcal{R}_2^1}\}$ and $\mathcal{C}_2 \equiv \{\mathcal{R}_2, \tilde{\mathcal{R}_1^1}, \tilde{\mathcal{R}_2^1}\}$ where the *confidence* of each association in the CAC are $\mathcal{C}_1$:{1,0.2,0.8} and $\mathcal{C}_2$:{1,0.5,0.55}. Using the measure defined by Formula 6, $\mathcal{C}_1$ scores higher than $\mathcal{C}_2$ (0.5 $>$ 0.475). However, intuitively the contextual association in $\mathcal{C}_1$ with 0.8 *confidence* indicates that the ADRs are more likely to be caused by one of the individual drugs. In this example, $\mathcal{C}_2$ should score higher since all of its contextual associations have relatively lower *confidence* as compared to the target association. To overcome this, we now introduce the coefficient of variation to penalize the CAC with diverse contextual associations w.r.t their *confidence*:

$$contrast_{cv}(\mathcal{C}) = contrast_{avg}(\mathcal{C}) \times G(\mathcal{C} - \mathcal{R}), \tag{7}$$

where

$$G(\mathcal{S}) = (1 - \theta \cdot C_v(\mathcal{S})), \tag{8}$$

$C_v(\mathcal{S})$ computes the coefficient of variation of the *confidence* set of a set of associations $\mathcal{S}$, while θ denotes a user-specified parameter $(0 \leq \theta \leq 1)$ that controls the effect of this penalty. Using the previous example with $\theta = 0.75$, then $contrast_{cv}(\mathcal{C}_1)$ = 0.18 and $contrast_{cv}(\mathcal{C}_2)$ = 0.45 where *contrast*$_{cv}$ is in favor of $\mathcal{C}_2$ now.

A drug-safety evaluator is typically knowledgeable about the individual drugs but may be less experienced with unknown MDARs. To expose more complicated cases, MARAS assigns more weight to the contextual association with less drugs. For example, if there are 3 drugs in the target association, the weak association between each individual drug and the ADRs is more important than the weak association between any 2 of the drugs and the ADRs. By considering this, the CAC that involves more drugs should get higher score so that it is pointed out to the drug-safety evaluator. Therefore, we design the final *contrast* score as below:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}(\mathcal{P}_c(\mathcal{R}) - \mathcal{P}_c(\tilde{\mathcal{R}}_j^i)) \times H(i,n) \times G(\{\tilde{\mathcal{R}}^i\}), \tag{9}$$

where $H(i,n)$ is a weighting function that is inversely proportional to the number of drugs in an association, i the number of drugs in $\tilde{\mathcal{R}}_j^i$, n the number of drugs in $\mathcal{R}$, and $\{\tilde{\mathcal{R}}^i\}$ denotes the set of contextual associations with the same number of drugs (i). In our experiment, $H(i,n)$ is chosen to be a linear decay function where $H(i,n) = (1-(i-1)/n)$, though other functions are possible.

2.4 Interactive Temporal Association Analytics

We now introduce our **TARA** model framework for interactive exploration of associations from evolving data.

2.4.1 Time Dimension of the TARA Model

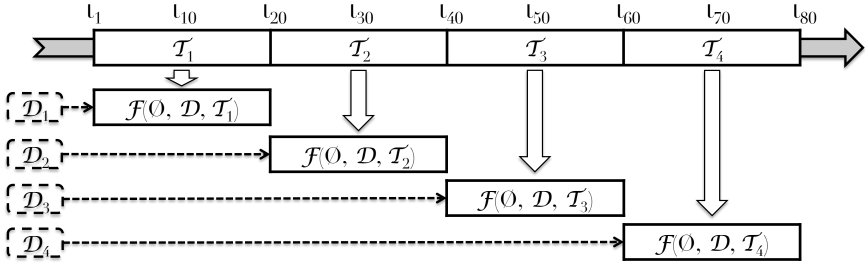

t_1
t_{10}
t_{20}
t_{30}
t_{40}
t_{50}
t_{60}
t_{70}
t_{80}
$\mathcal{T}_1$
$\mathcal{T}_2$
$\mathcal{T}_3$
$\mathcal{T}_4$
$\mathcal{D}_1$
$\mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_1)$
$\mathcal{D}_2$
$\mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_2)$
$\mathcal{D}_3$
$\mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_3)$
$\mathcal{D}_4$
$\mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_4)$

Figure 3: Tumbling Window Model of TARA

Data analysts often are interested in exploring the associations that hold in particular time periods, such as an hour or a day. Coarser time specifications can be broken down to ranges of smaller granularities. Moreover, the measures of an association in a longer time period can then be computed based on the measures of the associations in the shorter periods that are part of this longer period. Based on this observation, **TARA** partitions the data set into disjoint time periods, called *windows*. Mining queries with a coarser time

Table 1: Example of pregenerated temporal association rule.

Itemset	Support	
	$\mathcal{T}_1$	$\mathcal{T}_2$
a	0.36	0.44
b	0.45	0.22
c	0.36	0.44
ab	0.18	0.11
ac	0.18	0.33
bc	0.09	0.11

(a) Itemset (min supp = 0.05).

Rule	(Support, Confidence)	
	$\mathcal{T}_1$	$\mathcal{T}_2$
$\mathcal{R}_1$: a->b	(0.18, 0.5)	(0.11, 0.25)
$\mathcal{R}_2$: b->a	(0.18, 0.4)	(0.11, 0.5)
$\mathcal{R}_3$: a->c	(0.18, 0.5)	(0.33, 0.75)
$\mathcal{R}_4$: c->a	(0.18, 0.5)	(0.33, 0.75)
$\mathcal{R}_5$: c->b	(0.09, 0.25)	(0.11, 0.25)
$\mathcal{R}_6$: b->c	-	(0.11, 0.5)

(b) Rule (min conf = 0.25).

granularity settings than this basic *window size* are then supported using roll-up operations.

Let $\mathcal{D}$ be the evolving data set and w be the basic window size that represents the minimum granularity. Therefore, the set of *times* $\mathcal{T}$ contains disjoint but consecutive time periods each of size w denoted by $\mathcal{T} = \{..., \mathcal{T}_i, ..., \mathcal{T}_j, ...\}$, $(\forall \mathcal{T}_i, \mathcal{T}_j)$, if $\mathcal{T}_i \neq \mathcal{T}_j$, and $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$. The evolving data set $\mathcal{D}$ is partitioned into small chunks according to each time period $\mathcal{T}_i$ in $\mathcal{T}$ denoted by $\mathcal{D} = \{..., \mathcal{D}_i, ..., \mathcal{D}_j, ...\}$ where $\mathcal{D}_i = \mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_i)$. In Figure 3, for example we set the *window size* $w = 20$. That is, the time frame is partitioned into a set of time periods of length 20, e.g $\mathcal{T}_2 = [t_{21}, t_{40}]$. The evolving data set $\mathcal{D}$ is partitioned into time-oriented data subsets $\mathcal{D}_i$ according to these time periods, e.g. $\mathcal{D}_2 = \mathcal{F}(\emptyset, \mathcal{D}, \mathcal{T}_2)$. For each data partition $\mathcal{D}_i$, **TARA** pregenerates the associations off-line. **TARA** processes the raw data $\mathcal{D}$ once to pregenerate the temporal associations held in these windows. A query with the coarser time specification can then be answered based on these pregenerated associations.

Definition 8. ***Time availability***: *Let w be the finest time granularity. Then $\mathcal{T}^w = \{..., \mathcal{T}_i^w, ..., \mathcal{T}_j^w, ...\}$ corresponds to the basic time periods of $\mathcal{T}$ that are generated by* ***TARA*** *through time partitioning by w. A time specification $\mathcal{T}_k$ supported in* ***TARA*** *thus is $\mathcal{T}_k = \bigcup_{m=i}^{j} \mathcal{T}_m^w$, where $i \leq j$.*

This strategy allows us to support **roll-up** and **drill-down** of time periods at run time such as days, months or years to support long and short term goals.

2.4.2 Evolving Parameter Space Model

In association rule mining, the input parameter values of *minimum support* and *confidence* can be any real number within [0,1]. Each combination, referred to as **parameter setting**, corresponds to a set of rules generated by using this parameter setting. We now extend this into the notion of an *Evolving Parameter Space* (*EPS*) that models relationships and distribution of rules across the multi-dimensional temporal parameter space.

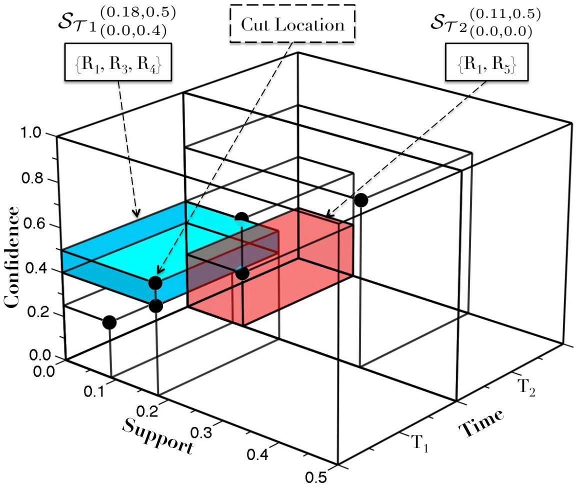


Figure 4: Evolving Parameter Space

Definition 9. ***Evolving Parameter Space**: Let $\mathcal{D}$ be an evolving data set, $\mathcal{D}_i$ be a **partition** of $\mathcal{D}$ by a basic time granularity $\mathcal{T}_i$, $\forall \mathcal{T}_i \in \mathcal{T}$. Let p_j be one of the n parameters. The (n+1)-dimensional space, denoted by $\mathcal{E}$ = { p_1,...,p_n, $\mathcal{T}$} and called **Evolving Parameter Space (EPS)**, organizes the rules $\{\mathcal{R}\}^{\mathcal{T}}$ where $\{\mathcal{R}\}^{\mathcal{T}} = \bigcup_{i=1}^{k}\{\mathcal{R}\}^{\mathcal{T}_i}$ and k is the total number of time partitions of $\mathcal{D}$. A temporal association rule $\mathcal{R}$ is associated with its **temporal parametric locations** ($\mathcal{R}$.value(p_1),..., $\mathcal{R}$.value(p_n))$^{\mathcal{T}_i}$ where $\mathcal{R}$.value(p_j) denotes the value of the j^{th} parameter for rule $\mathcal{R}$ in time $\mathcal{T}_i$.*

For simplicity, we use two parameters, namely *support* and *confidence* while others could be defined as well. Thus henceforth, the *EPS* $\mathcal{E}$ is a 3-dimensional space with *support*, *confidence* and *time* as its dimensions. A *temporal parametric location* depicting a rule $\mathcal{R}$ in time $\mathcal{T}_i$, denoted as $\mathcal{R}(supp, conf)^{\mathcal{T}_i}$, is represented as a line segment indicating the parameter values of $\mathcal{R}$ in $\mathcal{T}_i$. Rules $\mathcal{R}_1$, $\mathcal{R}_3$ and $\mathcal{R}_4$ map to the same *temporal parametric location* $(0.18, 0.5)^{\mathcal{T}_1}$ in the time period $\mathcal{T}_1$. However in time $\mathcal{T}_2$, $\mathcal{R}_1$ travels in the space so that now it maps to same location as $\mathcal{R}_5(0.11, 0.5)^{\mathcal{T}_2}$.

Lemma 2. *Let $\mathcal{L}$ denote a temporal parametric location in the EPS $\mathcal{E}$, $\mathcal{L}.p_i$ be the value of parameter p_i for location $\mathcal{L}$. Given a set of temporal parametric locations in the same time period $\mathcal{T}_i$, $\forall \mathcal{L}_m, \mathcal{L}_n \in \{\mathcal{L}\}$, where $m \neq n$, if there exists a p_i such that $\mathcal{L}_m.p_i \neq \mathcal{L}_n.p_i$, then the temporal association rules that map to $\mathcal{L}_m$ are guaranteed to be distinct from those that map to $\mathcal{L}_n$.*

Proof. Rules' *temporal parametric locations* in time $\mathcal{T}_i$ are generated from the same data partition $\mathcal{D}_i$. Any given rule at time $\mathcal{T}_i$ cannot have two distinct values for one parameter.

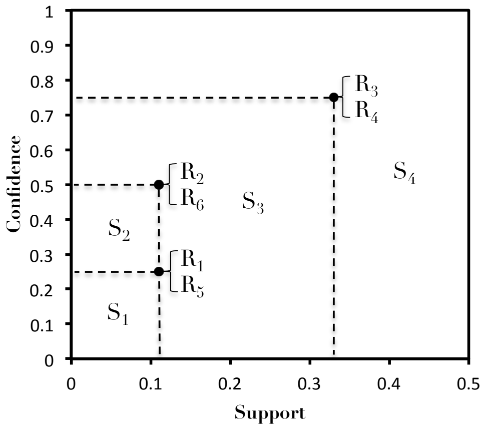


Figure 5: Evolving Parameter Space slice at Time $\mathcal{T}_2$

Therefore, a rule $\mathcal{R}$ cannot map to two distinct *temporal parametric locations* within the same time. □

Each rule's *temporal parametric location* can either remain steady or change over multiple time periods. We call this stream of locations the **trajectory of the association**.

Definition 10. ***Trajectory of an association***: *Given a sequence of time periods* $\{\mathcal{T}\} = \{\mathcal{T}_1..., \mathcal{T}_m\}$*, the* ***trajectory of an association*** $\mathcal{R}$ *in* $\{\mathcal{T}\}$ *is the set of temporal parametric locations that represent its parameter values in the time periods in* $\{\mathcal{T}\}$*.*

This trajectory of a rule allows us to compute different measures about the rule that summarize its evolving patterns like *coverage* [95], *stability* [67] and *standard deviation*. These measures can be computed for each individual rule or even for a set of rules to provide individual or global summarization respectively.

Given a data set with n unique items, the maximum number of rules is finite, bounded by $3^n - 2^n + 1$ [66]. Therefore, some set of parameter settings must correspond to same set of rules. Figure 5 shows a slice of the evolving parameter space for time $\mathcal{T}_2$. Rules with identical parameter values are represented by the same point in this space. These points partition the space into 4 regions marked by dashed lines. If a user specified *minimum support* and *confidence* configuration for mining falls into region $\mathcal{S}_3$, then regardless of its actual position within the region, the output ruleset is always $\{\mathcal{R}_3, \mathcal{R}_4\}$. This observation is inspired by the work presented in [66]. Thus the entire evolving parameter space at a time $\mathcal{T}_i$ can be partitioned into a finite set of regions referred to as *time-aware stable regions*. This notion of *time-aware stable regions* forms our coarse granularity abstraction of the temporal association rules generated from an evolving data set $\mathcal{D}$.

Definition 11. ***Time-Aware Stable Regions***: *Given an EPS* $\mathcal{E}$ *of n parameters* $\{p_1,...,p_n\}$ *and times* $\mathcal{T}$ *as* $(n+1)$ *dimensions for an evolving data set* $\mathcal{D}$*, then a* ***time-aware stable region*** *in a* *time period $\mathcal{T}_i$ is a closed hyper-box denoted by $\mathcal{S}_{\mathcal{T}_i}{}^{(upper(p_1),...upper(p_n))}_{\{(lower(p_1),...lower(p_n))\}}$ with its boundary specified by locations ($\mathcal{S}.upper(p_1)$,...,$\mathcal{S}.upper(p_n))^{\mathcal{T}_i}$ and $\{(\mathcal{S}\ .lower(p_1)$,...,$\mathcal{S}.lower(p_n))^{\mathcal{T}_i}\}$ within each of which no matter how the parameter values are adjusted, the set of rules generated from $\mathcal{D}_i$ remains unchanged.*

Considering the 3-dimensional *EPS* in Figure 4, a *time-aware stable region* is bounded by an upper location $(supp_u$,$conf_u)^{\mathcal{T}_i}$ and k lower locations $\{(supp_{l_j}$,$conf_{l_j})^{\mathcal{T}_i}\}$ where $j \in [1, k]$. The *support* and *confidence* values of the upper location will always be greater than those of all its lower points, i.e., $\forall j$ $(supp_u \geq supp_{l_j})$ and $(conf_u \geq conf_{l_j})$. The upper location of a *time-aware stable region* is called its **cut location**.

Definition 12. ***Cut Location**: Let EPS $\mathcal{E}$ be a 3-dimensi-onal space with support x, confidence y and time z as its dimensions, $\{\mathcal{X}\}$ be a set of the intersections formed by the perpendicular projections of each temporal parametric location onto x and y planes. The cut locations within $\mathcal{E}$ are then denoted by $\{\mathcal{C}\}$, where $\{\mathcal{X}\}$ = $\{\mathcal{C}\} \cup \{\mathcal{L}\}$.*

Figure 4 depicts *time-aware stable regions* $\mathcal{S}_{\mathcal{T}_1}{}^{(0.18,0.5)}_{(0,0.4)}$ and $\mathcal{S}_{\mathcal{T}_2}{}^{(0.11,0.5)}_{(0,0)}$. For region $\mathcal{S}_{\mathcal{T}1}{}^{(0.18,0.5)}_{(0,0.4)}$, the *cut location* is (0.18, 0.5)$^{\mathcal{T}_1}$. It is bounded by the parametric locations (0.18,0.5)$^{\mathcal{T}_1}$ and (0,0.4)$^{\mathcal{T}_1}$ and contains rules $\mathcal{R}_1$, $\mathcal{R}_3$ and $\mathcal{R}_4$.

Lemma 3. *Given a set of time-aware stable regions $\{\mathcal{S}\}$ for the same $\mathcal{T}_i$, $\forall$ $\mathcal{S}_m$, $\mathcal{S}_n \in \{\mathcal{S}\}$, where $m \neq n$, the associations that map to the cut location of $\mathcal{S}_m$ are guaranteed to be distinct from the ones that map to the cut location of $\mathcal{S}_n$.*

Proof. By Lemma 2, rules generated in the same time period but map to different *temporal parametric locations* are guaranteed to be distinct. The locations in $\{\mathcal{X}\}$ either belong to $\{\mathcal{L}\}$ or have no rule. Therefore, within a time period $\mathcal{T}_i$, rules that map to different *time-aware stable regions* are guaranteed to be distinct. □

Definition 13. ***Dominating Stable Region**: A time-aware stable region $\mathcal{S}_m$ **dominates** region $\mathcal{S}_n$ where $m \neq n$, if and only if $\forall p_i \in \mathcal{P}$ $\mathcal{S}_m.\mathcal{C}.p_i \leq \mathcal{S}_n.\mathcal{C}.p_i$, and $\mathcal{S}_m$ and $\mathcal{S}_n$ are in same $\mathcal{T}_i$ where $\mathcal{S}_m.\mathcal{C}$ refers to the cut location of stable region $\mathcal{S}_m$.*

Lemma 4. *Considering two time-aware stable regions $\mathcal{S}_m$ and $\mathcal{S}_n$ where $m \neq n$. If $\mathcal{S}_m$ dominates $\mathcal{S}_n$, then rules valid within the dominated region $\mathcal{S}_n$ are also valid in the dominating region $\mathcal{S}_m$ but not vice versa.*

Proof. A temporal rule $\mathcal{R}_i$ is in the final output ruleset if in the specified $\mathcal{T}_k$, $\forall$ p_j, $\mathcal{R}_i$.value(p_j) $\geq$ $min\ parameters$ where $p_j \in \{p_1$,...,$p_n\}$. If $\mathcal{R}_i$ belongs to region $\mathcal{S}_n$, the *temporal parametric*

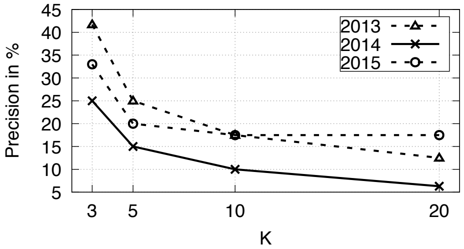


Figure 6: Precision of top K MARAS MDAR signals.

location of $\mathcal{R}_i$ is the upper location of $\mathcal{S}_n$. Because $\mathcal{S}_m$ dominates $\mathcal{S}_n$, $\forall$ p_j, $\mathcal{S}_m.upper(p_j)$ $\leq \mathcal{S}_n.upper(p_j)$, meaning $\forall$ p_j, $\mathcal{S}_m.upper(p_j) \leq \mathcal{R}_i$.value($p_j$). So $\mathcal{R}_i$ is valid in $\mathcal{S}_m$ as well. However, vice versa is not true, as can be trivially shown. □

Consider $\mathcal{S}_1$ = $\mathcal{S}_{\mathcal{T}_1}{}^{(0.18,0.5)}_{(0,0.4)}$ and $\mathcal{S}_2$ = $\mathcal{S}_{\mathcal{T}_1}{}^{(0.09,0.25)}_{(0,0)}$ in Figure 4. Based on Def. 13, $\mathcal{S}_2$ dominates $\mathcal{S}_1$ because every parameter value in the upper location of $\mathcal{S}_2$ is smaller than the corresponding value of $\mathcal{S}_1$.

If the rules in region $\mathcal{S}_{\mathcal{T}_1}{}^{(0.09,0.25)}_{(0,0)}$ are included in the final result, then region must also contain the rules that are valid in $\mathcal{S}_{\mathcal{T}_1}{}^{(0.18,0.5)}_{(0,0.4)}$. By Lemma 4, given a user-specified parameter setting, once a region is identified as a valid region to produce the final ruleset, all its dominated regions should then also be included in the user output.

Using this concept of dominating stable regions [66], each rule is stored once in the stable region and by iterating over its dominating regions the final ruleset can simply be obtained for a given pair of *support* and *confidence* values.

2.5 Experimental Results

2.5.1 Effectiveness of the Contrast Measure

The FAERS Data Source. We work with ADR reports from FAERS, a reporting system and database maintained by the FDA as a part of its post-marketing drug safety surveillance program. It contains million of records about adverse events and medication errors. To ensure the **reproducibility** of this experiment, we used the public version of the FAERS [3] data available quarterly from 2013-15. We selected the mandatory reports submitted by manufacturers marked as expedited (EXP). Each quarter has 100k - 160k reports, 30k - 37k reported drugs and 9k - 10k reported ADRs.

Quality of MDAR Signal. The main purpose of MARAS is to alert the drug-safety reviewers about possibly unknown MDAR cases collected through the post-market surveillance programs. There is no benchmark database that can be used to systematically evaluate how one should most effectively signal MDARs using ADR reports i.e., no "golden standard". Therefore, one of our evaluation strategies is to evaluate the effectiveness of MARAS by measuring the precision in terms of a hit of a known MDARs. The two sources we used are Drugs.com [1], a FDA recommended resource for obtaining information on known MDARs and DrugBank [35], a drug database that contains comprehensive biochemical and pharmacological information providing insights on MDARs. Figure 6 shows the precision of MARAS within the top k results. Precision is defined by the ratio of the number of hits to the number of the signals. "Precision at K" measures the accuracy of MARAS for signaling the known MDAR as well as the effectiveness of the *contrast* measure for ranking the returned signals. The precision of MARAS for each year is an average precision on 4 quarters data. There are relatively more hits in the higher ranked results, thus proving the effectiveness of our ranking strategy.

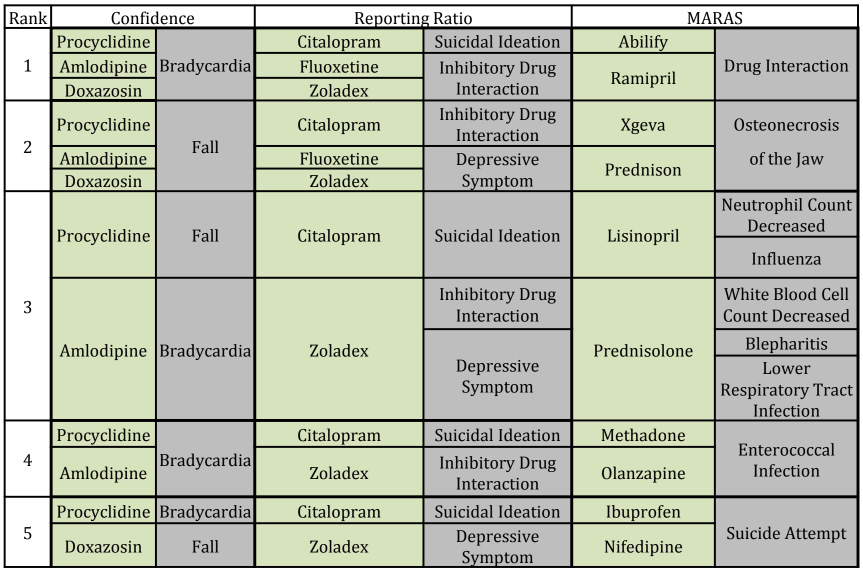
Case Study. Here, we report a case study on three top signals detected by MARAS. The goal of our case study using FAERS ADR reports is to validate the top ranked MDARs identified by MARAS through domain knowledge resources.

Case I: Eliquis and Ibuprofen (Detected and ranked 2^{nd} by MARAS in 2014-Q2 dataset). *Eliquis (Apixaban)*, an anticoagulant for the treatment of venous thromboembolic events is used to prevent platelets in the blood from sticking together and forming a blood clot. *Ibuprofen* is a nonsteroidal anti-inflammatory drug used to reduce inflammation and pain in the body. According to Drugs.com and DrugBank, using these two drugs together may increase the anticoagulant activities of *Apixaban*, lowering the body's ability to form clots and may cause increased bleeding, including severe and sometimes fatal hemorrhage.

Case II: Ondansetron and Lithium (Detected and ranked 1^{st} by MARAS in 2014-Q3 dataset). *Ondansetron* is used to prevent nausea and vomiting that may be caused by surgery or by medicine to treat cancer. *Lithium* is used to treat the manic episodes of bipolar disorder. According to DrugBank, "*Lithium* may increase the neurotoxic activities of *Ondansetron*". Neurotoxicity occurs when the exposure to natural or man-made toxic substances (neurotoxicants) alters the normal activity of the nervous system [2]. According to Drugs.com, "using the two drugs together can increase the risk of a rare but serious condition called the serotonin syndrome, which may include symptoms such as confusion, hallucination, seizure, extreme changes in blood pressure, increased heart rate, fever. Severe cases may result in coma and even death".

Case III: Abilify and Ramipril (Detected and ranked 1^{st} by MARAS in 2015-Q3 dataset). *Abilify (Aripiprazole)*, an antipsychotic medication is used to treat the symptoms of psychotic conditions such as schizophrenia and bipolar I disorder. *Ramipril*, an ACE inhibitor is used to treat high blood pressure or congestive heart failure. According to Drugs.com and DrugBank, these two medications taken in combination can have an additive effect in lowering blood pressure and can cause headache, dizziness, fainting, and/or changes in pulse or heart rate.

Table 2: Top 5 MDAR signals from 3^{rd} Quarter of 2015.



Rank	Confidence		Reporting Ratio		MARAS	
1	Procyclidine	Bradycardia	Citalopram	Suicidal Ideation	Abilify	Drug Interaction
	Amlodipine		Fluoxetine	Inhibitory Drug Interaction	Ramipril	
	Doxazosin		Zoladex			
2	Procyclidine	Fall	Citalopram	Inhibitory Drug Interaction	Xgeva	Osteonecrosis of the Jaw
	Amlodipine		Fluoxetine	Depressive Symptom	Prednison	
	Doxazosin		Zoladex			
3	Procyclidine	Fall	Citalopram	Suicidal Ideation	Lisinopril	Neutrophil Count Decreased
						Influenza
	Amlodipine	Bradycardia	Zoladex	Inhibitory Drug Interaction	Prednisolone	White Blood Cell Count Decreased
				Depressive Symptom		Blepharitis
						Lower Respiratory Tract Infection
4	Procyclidine	Bradycardia	Citalopram	Suicidal Ideation	Methadone	Enterococcal Infection
	Amlodipine		Zoladex	Inhibitory Drug Interaction	Olanzapine	
5	Procyclidine	Bradycardia	Citalopram	Suicidal Ideation	Ibuprofen	Suicide Attempt
	Doxazosin	Fall	Zoladex	Depressive Symptom	Nifedipine	

Comparison to State-of-the-Art Baselines.

Table 2 shows top 5 MDAR signals generated each from 2015 Q3 data by three different methods namely *Confidence* [115], *Reporting Ratio* [43] (Lift) and MARAS as depicted in the columns one, two and three respectively. The first two columns show the associations between drugs and ADRs ranked by their *confidence* and *RR* values respectively. These two methods do not filter spurious associations. As a result, there are many similar redundant and possibly misleading signals.

In contrast, top ranked signals generated by MARAS are more diverse as compared to those produced by the first two methods. Worse yet, the top ranked signals produced by MARAS signals on interaction between *Rampiril* and *Abilify* as verified via a case study is ranked 2,436th by *confidence* and 16,984th by *RR*. Similarly, the second top ranked association by MARAS that shows interaction between *Xgeva* and *Prednison* can lead to osteonecrosis of jaw is ranked 2,166th by *confidence* and 9,312th by *RR*. Thus by using the *Confidence* or *Reporting Ratio* (RR) we would risk important findings staying hidden in the association set. Hence we can deduce that MARAS successfully detects non-spurious and non-redundant MDARs, which other methods fail to detect.

2.5.2 Efficiency of TARA

Experimental Setup. Experiments are conducted on a OS X machine with 2.4 GHz Intel Core i5 processor and 8 GB RAM. The system and its competitors are implemented in C++ using Qt Creator with Clang 64-bit compiler.

Table 3: Datasets

	100retail	*T5k*	*T2k*	*webdocs*
Transactions	8,816,200	5,000,000	2,000,000	1,692,082
Unique Items	16,470	23,870	30,551	5,267,656
Avg Len of Tran	10	50	100	177
Size	416.8 MB	1.48 GB	1.38 GB	1.48 GB

Datasets. We select a variety of datasets with diverse characteristics here. The benchmark datasets, *T5kL50N100* and *T2kL100N1k*, are generated by the *IBM Quest data generator* [5] modeling transactions in a retail store. We partition these datasets into 5 equal-sized batches to form the evolving data sources. The *retail* dataset [19] contains 88,163 transactions collected from a Belgian retail supermarket store in 5 months. To study scalability, we replicate this *retail* dataset 100 times. The *webdocs* dataset [70] is built from a spidered collection of web html documents. Both of these real datasets are partitioned into 10 equal-sized batches to form evolving data sources.The statistics of the datasets are summarized in Table 3.

Alternate State-of-the-art Techniques. The performance of **TARA** is compared against three competitors. **DCTAR** [65] derives the ruleset directly from the raw data given a parameter configuration. It computes the associations from scratch whenever a new batch of data arrives. **H-Mine** [111] instead pregenerates the intermediate frequent item sets offline. For specific parameter settings, the algorithm utilizes the itemsets to generate the associations online instead of extracting them from the raw data. **PARAS** [66] pregenerates frequent itemsets and rules offline for the entire data set assuming all data is static and given apriori. That is, time is ignored. For our experiments, we construct the PARAS index for a single time period. However at online time if request comes for different periods it then generates the associations from scratch.

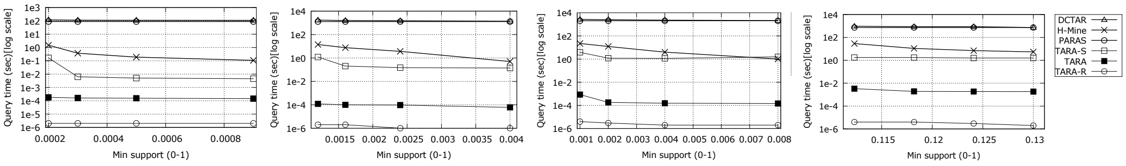
Experimental Methodologies: The performance of our approach and state-of-the-art algorithms is measured by:

Offline Preprocessing Time. We measure the single and multiple data batches preprocessing time for **TARA**, H-Mine and PARAS. Since DCTAR does not involve any preprocessing, it is excluded from this measurement.

Online Processing Time. We measure the online processing time for a query averaged over multiple runs to evaluate the speedup.

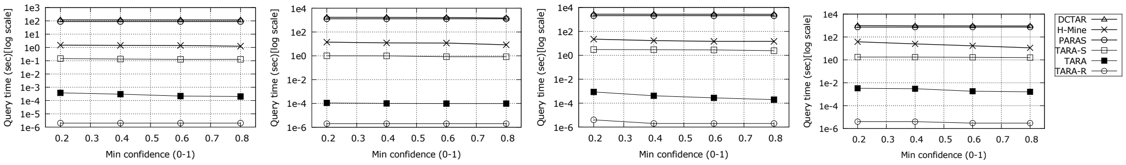
Size of Pregenerated Information. We compare the sizes of the preprocessed information. DCTAR is again excluded. The size of the tree structure in H-Mine and the size of the *TAR Archive* in **TARA** are thus compared.

Evaluation of Preprocessing Time. We first compare the preprocessing times for H-Mine, PAR-AS and **TARA**. In the offline step, as the window slides, H-Mine (1) precomputes the frequent item sets and (2) stores them along with their associated *support* value



(a) retail (b) T5kL50N100 (c) T2kL100N1k (d) webdocs

Figure 7: Rule Trajectory and Parameter Recommendation: Varying Support



(a) retail (b) T5kL50N100 (c) T2kL100N1k (d) webdocs

Figure 8: Rule Trajectory and Parameter Recommendation: Varying Confidence

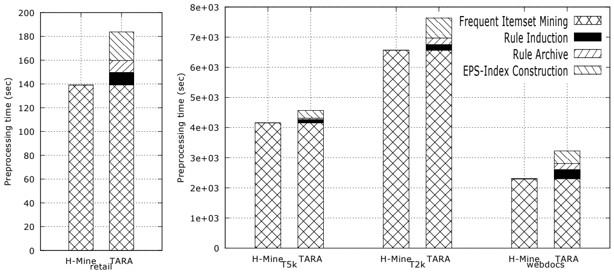


Figure 9: Preprocessing Time

Table 4: Thresholds for Indexes

Dataset	H-Mine	TARA&PARAS (supp, conf)
retail	0.0002	(0.0002, 0.1)
T5k	0.0012	(0.0012, 0.2)
T2k	0.001	(0.001, 0.2)
webdocs	0.1123	(0.1123, 0.2)

into a tree structure. Whereas **TARA** (1) precomputes the frequent item sets, (2) derives the ruleset, (3) archives them along with the associated *support* and *confidence* values and (4) updates the *EPS*-Index. PARAS proceeds with the same process as **TARA** except that it does not utilize the archive nor does it keep the pregenerated information from the previous windows. Therefore, the total preprocessing time of PARAS is similar to **TARA** except for the archival time. Figure 9 compares the preprocessing time of H-Mine and **TARA** for all windows for the *retail*, *T5kL50N100*, *T2kL100N1k* datasets, with the system threshold settings summarized in Table 4. As shown, frequent item set generation occupies a relatively large portion of the preprocessing time as compared to other tasks. This confirms prior works [41] that rule generation is more efficient compared to frequent itemset generation. Overall, the additional preprocessing tasks in **TARA** require no more than 20% extra time than H-Mine. This extra time gives significant advantage to **TARA** in terms of truly interactive online performance and support of many advanced exploration operations.

Evaluation of Online Processing Time. Next, we compare the online processing times (y-axis in log scale) for our proposed operations. The user-specified parameters, namely *minsupp*, *minconf* and *time periods*, are varied. The examined queries fall into two categories: (1) Rule trajectory and parameter recommendation queries and (2) Ruleset comparison queries. In the first experiment, we test the performance of **TARA** against the three competitors using several query types, namely $Q1$ and $Q3$ in *single match* mode. Second, we use $Q2$ in *exact match* mode to test the performance of **TARA** against others. We choose $Q1$, $Q2$ and $Q3$ because they cover the major exploration operations and subroutines in the online processing phase.

2.5.3 Trajectory and Parameter Recommendation

To process $Q1$, the system needs to find the rules that satisfy *minsupp* and *conf* in a single *time period* and examine their parameter values in other specified time periods. For DCTAR, it mines the rules from the transactions that fall into the last window and examines their parameter values by processing the transactions that fall into the 3 previous windows. For PARAS, the process is identical except that the rules are retrieved from the PARAS index built based upon the newest window. For H-Mine, the rules are derived and examined by using its item set index.

Impact of Varying Support and Confidence. To determine the effect of *minsupp*, we conduct several experiments by fixing *minconf* to a constant value and varying the *minsupp* value. Figure 7 illustrate the query processing times for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets with fixed *minconf* 0.4, 0.2, 0.2 and 0.4, respectively.

We observe that, **TARA** consistently outperforms DCTAR and PARAS by 6,7,7 and 5 orders and H-Mine by 3, 4, 4 and 4 orders of magnitude for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets respectively. **TARA-S** stands for the implementation of **TARA** with the rule index inside each *time sensitive stable region* to support content based exploration ($Q5$). The merging of indexes when *dominated regions* are being collected incurs extra costs as compared to the TARA system without these rule indexes. Especially when the number of rules in the result is small, this extra cost results in similar or slower response time compared to H-Mine as shown in Figs. 7(b) and (c). The reason of the fast response of **TARA** is that it prepares sufficient amount of information in the offline stage, so that answering such queries is simply about searching the **TARA** index.

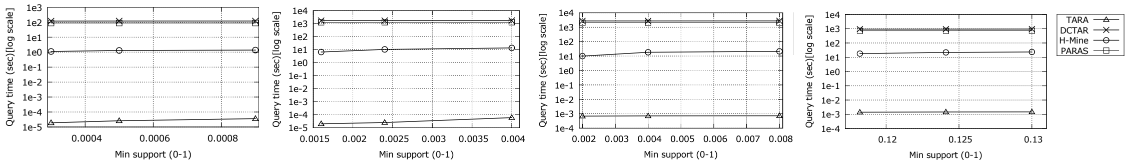
TARA-R shows the response time of returning the *time-sensitive stable region* which answers $Q3$. Since PARAS always builds the index for the latest window, in this particular experiment, it achieves the same response time as **TARA** because only regions that fall into the latest window are considered. All other systems are not capable of answering $Q3$. That is, using DCTAR and H-Mine, an analyst would need to generate all possible rules in the specified time period and then investigate all to find the answer.

Impact of Varying Confidence. Next, we fix the *minsupp* to a constant value and vary the *minconf* value. Figure 8 illustrates the query processing times for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets with fixed *minsupp* 0.0002, 0.0012, 0.0012 and 0.1123, respectively. Overall, both **TARA** and **TARA-S** consistently perform several orders of magnitude better than the three competitors.

2.5.4 Ruleset Comparison Queries

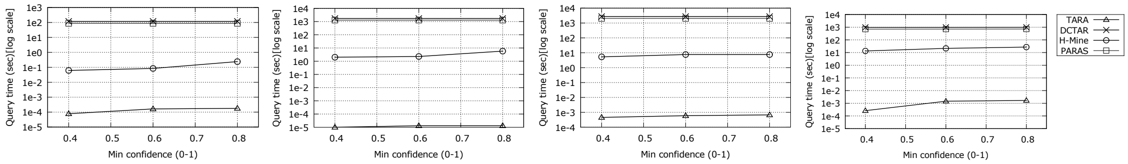
$Q2$ returns the differences of the rulesets w.r.t two parameter settings that share the same time specification. In this particular experiment, the query is configured with the *exact match* mode. It returns the differences of two parameter setting across 4 windows. Since the DCTAR and H-Mine do not support such query, we implement a subroutine in their rule derivation module to determine if the parameter value of the rule satisfies one setting but not the other. This subroutine is optimized so that it does not generate the overlapping ruleset w.r.t 2 different settings. In this experiment, we either fix *minsupp* or *minconf* and vary the other one.

Impact of Varying 2^{nd} Support. Figure 10 illustrates the query processing times for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets. The fixed *min parameters* for these datasets are ($minsupp_1$, $minconf_1$, $minconf_2$): (0.0002, 0.4, 0.4), (0.0012, 0.2, 0.2), (0.0012, 0.2, 0.2) and (0.1123, 0.4, 0.4), respectively. The query processing times increase with an increase in the *minsupp* because the increase of the deviation from $minsupp_1$ to $minsupp_2$ results in larger differences between the two parameter settings. In particular, **TARA** out-



(a) retail (b) T5kL50N100 (c) T2kL100N1k (d) webdocs

Figure 10: Ruleset Comparison: Varying 2^{nd} Support



(a) retail (b) T5kL50N100 (c) T2kL100N1k (d) webdocs

Figure 11: Ruleset Comparison: Varying 2^{nd} Confidence

performs DCTAR and PARAS by 6,7,6 and 6 orders, H-Mine by 4, 5, 4 and 4 orders for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets, respectively.

Impact of Varying 2^{nd} **Confidence**. Figure 11 illustrates the query processing times for *retail*, *T5kL50N100*, *T2kL100N1k* and *webdocs* datasets. The fixed *min parameters* for these four datasets are ($minsupp_1$, $minconf_1$, $minsu$
pp_2): (0.0002, 0.4, 0.0002), (0.0012, 0.2, 0.0012), (0.0012, 0.2, 0.0012) and (0.1123, 0.4, 0.1123), respectively. **TARA** consistently performed several orders of magnitude better than the three competitors.

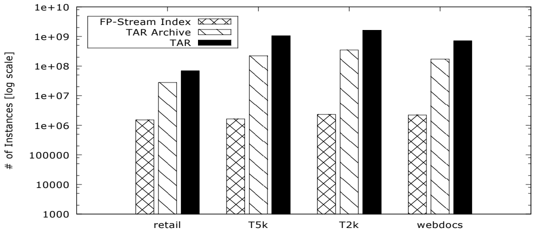


Figure 12: Size of the *TAR Archive*

Evaluation of Archive Size. We compare the sizes of the pregenerated information in **TARA**, H-Mine and PARAS. For H-Mine, the size of the structure is determined by the number of frequent item sets times the number of processed partitions, while the size of pre-stored information in **TARA** is determined by the size of the *TAR Archive*. PARAS only pregenerates the association in a single window. Its maximum size is $3^n - 2^n + 1$ where n is the unique items in that particular window. The actual index sizes can be estimated by multiplying the number of instances with the average space required per instance.

Figure 12 shows the size of the H-Mine Index, *TAR Archive* and the actual number of uncompressed rule parameter values for our four datasets with the system threshold settings summarized in Table 4. As **TARA** pre-generates rules instead of only the item sets, the size of the *TAR Archive* is larger than the H-Mine index. However, our encoding technique achieves favorable compression as compared to uncompressed rule parameter values.

2.6 Related Work

MDARs. [108, 109] used statistical methods to find interactions among drug classes. However, these methods are typically designed for a particular class of drugs or ADRs only. Hence, they do not consider all reported drugs and ADRs crucial for drug-surveillance. Unsupervised methods in particular association rule mining has been used in the medical domain to explore drug related ADRs [36, 51, 40]. These methods considered the identification of ADRs related to a single drug, rather than a combination of drugs.

ARL for Signaling MDAR. [43, 50] used ARL with *Reporting Ratio* (RR) and Proportional Reporting Ratio (PRR) respectively to find drug interactions triggering a set of ADRs. However, these approaches do not consider the association of individual drugs with the ADRs within a drug combination therefore providing many false positive signals. Cai et al [20] uses ARL and defines interestingness based on causal relation between two interacting drugs and ADRs. Moreover, none of these approaches remove spurious or misleading rules as introduced by our work.

Interestingness in ARL. Various attempts have been made in the literature to reduce the number of the generated rules and rank the most interesting ones [98, 120, 9]. However the majority of these measures are either for classification rules or are subjective measures that need domain specific knowledge to define interestingness. Sub-rules based interestingness has been studied by [31], where interestingness is defined as an unexpected confidence among a neighborhood. The interestingness based on sub-rule's confidence known as improvement [14] ensures that for every rule none of its simplifications offer any predictive advantage over it. None of these methods capture the most interesting associations among multiple drugs and ADRs.

Temporal association mining. Adding the time dimension in the context of association rules was first mentioned in [81]. However, while more follow-on works [38, 65, 111] improve the efficiency of temporal association mining by maintaining intermediate frequent item sets, all of these approaches require the user to input a specific parameter setting. This one-at-a-time approach not only limits efficiency, but also provides very limited feedback for the user.

Interestingness of temporal associations. [68, 97] identify the importance of analyzing the interestingness measures of associations. In the context of time-variant data, [67] measures the changes of the interestingness of the association w.r.t its histories. It is suggested that the interest in the rule itself is primarily determined by the interestingness of its change over time. Neither of these works tackle interactive mining through precomputation. In contrast, we explore the space of interestingness parameters for prestoring data mining results to facilitate fast online mining.

Interactive association mining. Prior works [22, 24, 66] have explored the space of parameters for handling data mining requests. However this work is restricted to static data. These approaches do not consider the time dimension as a property of the pattern. Instead we now study the problem of incorporating the time dimension into the association mining exploration process.

3 Temporal Local Outlier Detection

Manuscript

1. **Xiao Qin**, Lei Cao, Elke A. Rundensteiner and Samuel Madden. *Scalable Kernel Density Estimation-based Local Outlier Detection over Large Data Streams*. **EDBT**'19.

3.1 Introduction

Motivation. The phenomenal growth of digital devices coupled with their ever-increasing capabilities to generate and transmit live data presents an exciting new opportunity for real time data analytics. As the volume and velocity of data streams continue to grow, automated discovery of insights in such streaming data is critical. In particular, finding *outliers* in streaming data is a fundamental task in many online applications ranging from fraud detection, network intrusion monitoring to system fault analysis. In general, outliers are data points situated away from the majority of the points in the data space. For example, a transaction of a credit card in a physical location far away from where it has normally been used may indicate fraud. Over 15.4 million U.S residents were victims of such fraud in 2016 according to [6]. On the other hand, as more transactions take place in this new location, the previous transaction may appear legitimate as it begins to conform to the increasingly expected behavior exemplified by the new data. Thus, in streaming environments, it is critical to design a mechanism to efficiently identify outliers by monitoring the statistical properties of the data as it changes over time.

State-of-the-Art. To satisfy this need, several methods [87, 99] have been proposed in recent years that leverage the concept of local outlier [18] to detect outliers from data streams. Local outlier is based on the observation that real world datasets tend to be skewed, where different subspaces of the data exhibit different distribution properties. It is thus often more meaningful to decide on the outlier status of a point based on its difference from the points in its *local neighborhood* as opposed to using a global density [23] or frequency [8] cutoff threshold to detect outliers [33]. More specifically, a point x is considered a *local outlier* if the $\underline{\text{p}}$robability $\underline{\text{d}}$ensity (PD) at x is low *relative* to that at the points in x's local neighborhood.

Unfortunately, existing streaming local outlier solutions [87, 99] are not scalable to high volume data streams. The root cause is that they measure the probability density at each point x based on the point's distance to its k nearest neighbors (kNN). Unfortunately, kNN is very sensitive to data updates, meaning that the insertion or removal of even a small number of points can cause the kNN of many points in the dataset to be updated. Since the complexity of the kNN search [18] is quadratic in the number of the points, significant resources may be wasted on a large number of unnecessary kNN re-computations. Therefore, those approaches suffer from high response time when handling high-speed streams. For example, it took [87, 99] 10 minutes to processing just 100k tuples as shown in their experiments.

Intuitively, kernel density estimation (KDE) [102], an established probability density approximation method, could be leveraged for estimating the density at each point [107, 59, 100]. Unlike kNN-based density estimation that is sensitive to data changes, KDE estimates data density based on the statistical properties of the dataset. Therefore, it tends to be more robust to gradual data changes and is a better fit for streaming environments. However, surprisingly, to date no method has been proposed that utilizes KDE to tackle local outlier detection from data streams.

Challenges. Effectively leveraging KDE in the streaming context comes with challenges. First, the effectiveness of KDE depends on several factors. In particular, both the kernel function and the smoothing parameter (commonly referred to as *bandwidth*) [121] have to be carefully selected to achieve a high accuracy for density estimation. Further, to ensure the effectiveness of KDE in multimodal distributions prevalent in real world datasets, customized density estimators have to be established for different data subspaces. This raises the problem of how to select relevant kernel centers to enable the inference of these different estimators. Making correct decisions on all these factors is complex. Worst yet, the distribution characteristics of a data stream evolve. Therefore, these factors would have to be continuously tuned to fit the data.

Furthermore, similar to kNN search, the complexity of KDE is quadratic in the number of points [102]. While the computational costs can be reduced by running the density estimation on kernel centers sampled from the input dataset, sampling leads to a trade-off between accuracy and efficiency. Although a low sampling rate can dramatically reduce the computational complexity, one must be cautious because the estimated probability density at each point may be inaccurate due to an insufficient number of kernel centers. On the other hand, a higher sampling rate will certainly lead to a better estimation of the density. However, computational costs of KDE increase quadratically with more kernel centers. With a large number of kernel centers, KDE would be at risk of becoming too costly to satisfy the stringent response time requirements of streaming applications.

Due to the above challenges, to the best of our knowledge, no method has successfully adapted KDE to streaming data in an efficient manner to date.

Proposed Solution. In this work, we propose a KDE-based strategy for detecting top-N local outliers over streams, or in short **KELOS**. For the first time, KELOS makes local outlier detection practical to streaming data.

KELOS employs a new KDE-based semantics for streaming local outliers that focuses

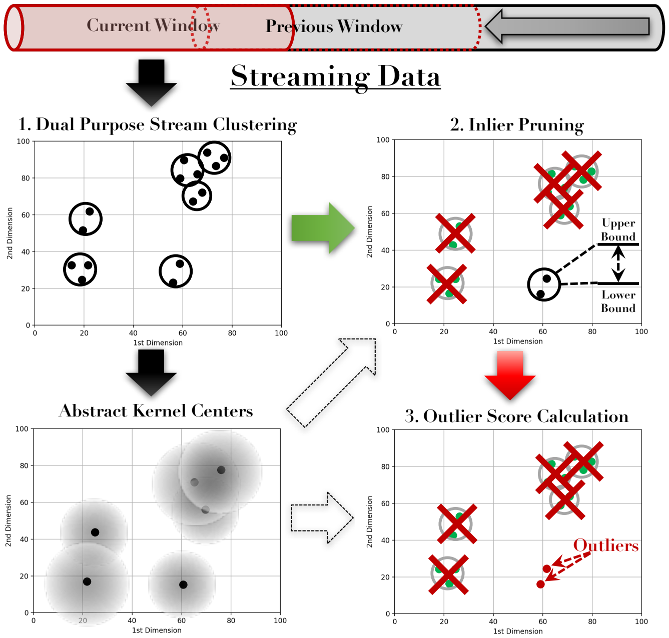


Figure 13: An illustration of KELOS approach.

on the continuous detection of the most promising outliers from data streams. Based on a thorough analysis of the properties of different kernel functions, we adopt the *product kernel function* for the semantics in KELOS. We show that this choice is appropriate for continuously approximating the density of multi-dimensional streaming data – key property needed for outlier detection. Furthermore, KELOS employs a data-driven bandwidth approximation mechanism that automatically adapts the bandwidth to the dynamics inherent in data streams. Thus, this choice of semantics establishes a promising foundation for the design of a scalable streaming local outlier detection method.

Second, KELOS solves the accuracy versus efficiency trade-off of KDE by introducing the notion of *abstract kernel centers*. The abstract kernel center concept is inspired by the nature of KDE, namely, in KDE, the density at a point x_i is determined by the *additive influences* from other points x_j, with the strength of the influence from one point x_j to another x_i being determined by the distance between x_i and x_j. As a result, points close to each other tend to have a similar influence on other points. These nearby points thus can be clustered together and considered as one abstract kernel center weighted by the data cardinality. Compared to the traditional sample point-based KDE, this strategy achieves higher accuracy in density estimation using *many fewer kernel centers*. Furthermore, although producing the abstract kernel centers typically is more expensive than sub-linear time complexity sampling, the small number of abstract kernel centers speeds up the later quadratic complexity process of local density estimation. This results in the overall computational costs for abstract kernel center-based KDE (aKDE) being much smaller than the traditional KDE, as shown in our experimental evaluation (Section 3.4.1).

Further, unlike existing techniques [87, 99], which detect outliers by routinely computing the probability density and then the outlierness score for every data point, KELOS employs an inlier pruning strategy. It quickly prunes the vast majority of the data points that have no chance to be outliers. The more expensive KDE method is only used afterwards to evaluate the remaining small number of individual potentially promising outlier candidates. Inlier pruning leverages the *stable density* observation that the data points in a tight cluster tend to share similar probability density and small outlierness scores. Moreover, the outlierness scores can be bounded based on the radius of the cluster.

Finally, inspired by micro-clustering [4] popular in the streaming context, KELOS uses a *dual purpose* stream clustering (DSC) approach to produce data clusters which are needed by both aKDE and the inlier pruning. By only doing a linear pass over the data, DSC not only produces the clusters, but also simultaneously collects the statistics sufficient for updating the density estimator of $aKDE$ and bounding the outlierness scores of each cluster for inlier pruning.

Putting all these optimizations together, we obtain the first linear time complexity streaming local outlier detection approach that thus scales to truly high speed streams as confirmed by our experiments.

Contributions. Our key contributions include:

- We propose new streaming local outlier detection semantics amenable for the design of scalable continuous local outlier detection strategies.
- We solve the effectiveness versus efficiency trade-off of KDE in the stream context by introducing the notion of abstract kernel centers. It by itself could be applied to a much broader class of density estimation related stream mining tasks beyond outlier detection.
- We propose a data-cluster granularity inlier pruning strategy that concentrates computation resources on strictly inspecting a small set of highly suspicious outlier candidates.
- We design a linear-time complexity data-clustering strategy that continuously produces the clusters yet collecting statistics sufficient for inlier pruning and density estimator updates.
- Our extensive experiments using public datasets with outlier labels demonstrate the effectiveness of KELOS in detecting outliers while achieving 3 orders of magnitude performance gain in computational costs against the alternative approaches.

3.2 Foundation

In this section we review the concepts of local outliers and kernel density estimation.

3.2.1 Local Outlier

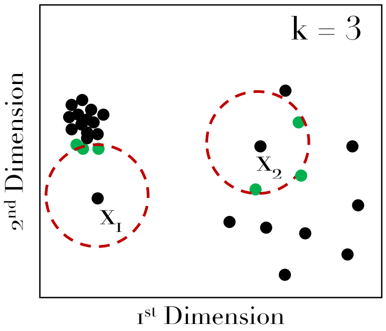


Figure 14: Local outlier detection using local densities.

The notion of local outliers [18] is based on the observation that different portions of a dataset may exhibit very different characteristics. It is thus often more meaningful to decide on the outlier status of a point based on its difference from the points in its local neighborhood as opposed to using a global density [23] or frequency [8] cutoff threshold to detect the outliers. Specifically, a point x_i is a local outlier if the density at x_i is significantly lower than the densities at x_i's neighbors.

As illustrated in Figure 14, although the densities at x_1 and x_2 are both low, the density at x_1 is quite different than the densities at the locations of its neighbors. However, the densities at the neighbors of x_2 is similar to x_2. Therefore, x_1 is more likely to be an outlier than x_2 due to its *relatively low density* in contrast to those at its neighbors.

Therefore, conceptually measuring a point x_i's status of being a local outlier corresponds to the two-steps:

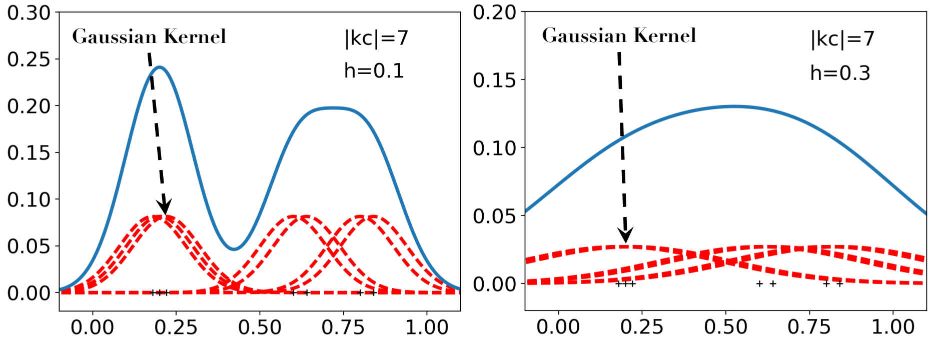
1. Estimate the density at x_i and the densities at its neighbors;
2. Compute the outlierness score of x_i based on the deviation of the density at x_i in contrast to those at its neighbors.

3.2.2 Kernel Density Estimation

Kernel density estimation (KDE) is a statistical method to estimate the probability density (PD) at the point in a dataset $X = \{x_1, \cdots, x_n\}$. Given a point $x_i \in X$, the kernel density estimator computes the density at x_i using a probability density function (PDF) $\tilde{f}(x_i)$:

$$\tilde{f}(x_i) = \frac{1}{m} \sum_{j=1}^{m} K_h(|x_i - kc_j|). \tag{10}$$

The core variables in Equation 10 are explained next.



(a) Bandwidth h = 0.1 (b) Bandwidth h = 0.3

Figure 15: An example of univariate kernel density estimator using Gaussian kernel with different bandwidth.

Kernel Centers. kc_j (1 $\leq$ j $\leq$ m) are called the kernel centers in the estimator. Typically, kc_j is a point sampled from X. The selected set of kernel centers should be sufficient to represent the data distribution of X [102].

Each kernel center kc_j carries a kernel function K_h. The *density contribution* by a kernel center kc_j is calculated based upon the distance from kc_j to the target point x_i. The density at x_i is estimated by the average density contribution by all kernel centers. For example, in Figure 15(a), there are 7 kernel centers. Each of them carries a kernel function (red dashed curve). The shape of the overall density function across all kernels is represented by the blue solid line. Given a dataset X with n points and m kernel centers, the time complexity of computing a density value at each and every point $x_i \in X$ is $\mathcal{O}(nm)$.

Kernel Function. A wide range of kernel functions can be used in kernel density estimation [102]. The most commonly used ones are the *Gaussian* and *Epanechnikov* kernel functions [33]:

$$K_{gauss}(u) = \frac{1}{(\sqrt{2\pi})h} e^{(-\frac{1}{2}\frac{u^2}{h^2})}, \tag{11}$$

$$K_{epanechnikov}(u) = \frac{3}{4h}\left(1 - \frac{u^2}{h^2}\right), \tag{12}$$

where u represents the distance from a kernel center kc_j to the target point x_i and h is an important smoothing factor, called *bandwidth*, explained below.

The bandwidth controls the smoothness of the shape of the estimated density function. The greater the value h, the smoother the shape of the density function $\tilde{f}$. As shown in Figures 15(a) and (b), using the same set of kernel centers but different bandwidth values, the estimated PDFs (the blue lines) are significant different from each other. Therefore, an appropriate bandwidth is critical to the accuracy of the density estimation.

Balloon Kernel for Modeling Local Density. In the context of local outlier detection, the balloon kernel is recommended [100] to handle multimodal distribution, where different subspaces of the data demonstrate different distribution properties. When estimating the density at a target point x_i, only the k nearest kernel centers of x_i are utilized in the estimator. Leveraging the study of *variable kernel density estimation* [110], this provides each point x_i a customized kernel density estimator that adapts to the distribution characteristics of x_i's surrounding area, hence also called *local density*. Selecting an appropriate k is critical and shown to be challenging [100].

3.3 KDE-based Local Outlier Detection from Data Streams

3.3.1 Density Estimator

In this section, we propose our abstract kernel center-based KDE strategy (aKDE). It solves the problem of accurately yet efficiently estimating the density at a given point. In contrast to the traditional sampling-based KDE approach [102], our density estimation is performed on top of a set of clusters (Figure 16) that succinctly summarize the distribution characteristics of the dataset. This approach is inspired by our *abstract kernel center* observation below.

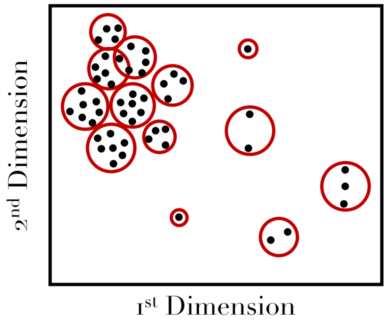
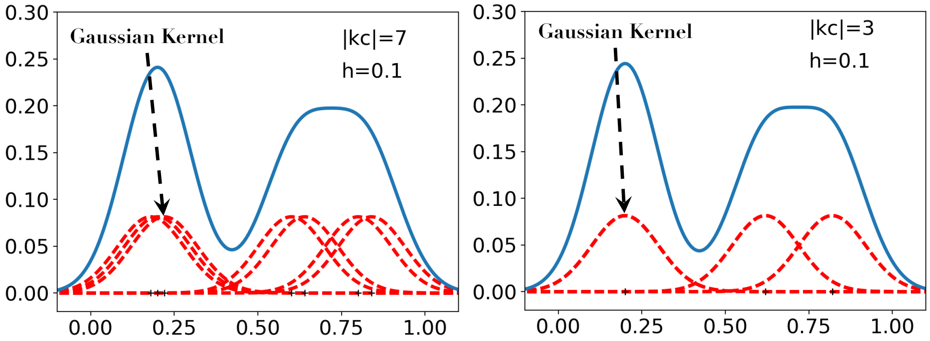


Figure 16: An illustration of abstract kernel centers (AKC). The abstract kernel centers are the virtual centroids of the clusters (red circle).

Abstract Kernel Center Observation. In KDE, the density at a given point x_i is determined by the *additive influences* of the kernel centers, while the influence from one center kc_j is determined by the distance between kc_j and x_i. The centers close to each other tend to have similar influence on the target point x_i. Therefore, clustering centers together and treating them as one abstract kernel center weighted by the cluster's data cardinality is effectively equivalent to using the whole set of data points as kernel centers.

Figure 17(b) shows an example estimation using the abstract kernel centers. The original 7 points in Figure 15(a) are abstracted into three clusters. The estimations (blue line) in Figure 17(b) with 3 centers and Figure 17(a) using all 7 points as kernel centers are similar.



(a) All points as kernel centers

(b) Abtract kernel centers

Figure 17: Local kernel density estimator.

On the performance side, real world data sets tend to be skewed. Therefore, typically most points can be clustered into a small number of tight clusters. Correspondingly the number of the abstract kernel centers tends to be much smaller than the number of sampled kernel centers that would be sufficient to represent the overall data distribution of the dataset. Since the core costs of local density estimation correspond to the computation of the k nearest kernel centers for each to be estimated point x_i, the small number of abstract kernel centers promises to reduce the complexity of the successive density estimation process.

Furthermore, the abstract kernel centers allow us to use a small k while establishing a diversified neighborhood – hence a comprehensive density estimator for each point. This not only reduces the complexity of the kNN search and kernel density computation, but also alleviates the problem of selecting an appropriate k. This selection, while critical for the accuracy of density estimation, is challenging as shown in the preliminaries (Section 3.2.2). Since the abstract kernel centers representing data clusters are more stable than sampled individual points in terms of their statistical properties, our selected k by such method would be more robust to the continuously changing stream data. Next, we formally define the concept of abstract kernel centers.

Definition 14. *Given a stream window $S^{W_c} = \{x_1, \cdots, x_n\}$, the abstract kernel centers of S^{W_c} are a set of pairs $\mathbb{AKC}(S^{W_c}) = \{\langle c_{c_1}, |c_1|\rangle, \cdots, \langle c_{c_m}, |c_m|\rangle\}$, where c_{c_i} $(1 \leq i \leq m)$ corresponds to the centroid of the respective data cluster c_i and $|c_i|$ the number of points in c_i. Here $\bigcup_{i=1}^{m} c_i = S^{W_c}$ and $\forall i, j, i \neq j$ $c_i \cap c_j = \emptyset$.*

Weighted Kernel Density Estimator. Intuitively, each abstract kernel center represents the centroid of a cluster of points close to each other along with the data cardinality of this cluster. Utilizing these abstract kernel centers, we construct a weighted kernel density estimator [39], where the kernel centers correspond to the centroids in $\mathbb{AKC}(S^{W_c})$ (the first component of $\mathbb{AKC}$) and the weight corresponds to the cardinality of the data cluster represented by the centroid (the second component). Therefore, the weighted kernel density estimator reflects the distribution characteristics of the entire dataset by utilizing only a small number of kernel centers. The formula is shown below:

$$\tilde{f}_{\mathbb{AKC}(S^{W_c})}(x_i) = \sum_{j=1}^{k} \omega(c_{c_j}) \prod_{l=1}^{d} K_{h^l}(|x_i^l - c_{c_j}^l|) \tag{13}$$

and

$$\omega(c_{c_j}) = \frac{|c_j|}{\sum_{m=1}^{k} |c_m|}, \tag{14}$$

where $c_{c_m} \in kNN(x_i, \mathbb{AKC}(S^{W_c}))$. Here $\tilde{f}_{\mathbb{AKC}(S^{W_c})}(x_i)$ in Equation 13 corresponds to a weighted product kernel estimator that computes the local density at x_i and $kNN(x_i, \mathbb{AKC}(S^{W_c}))$ corresponds to the k nearest centroids of x_i in the abstract kernel centers.

Bandwidth Estimation. One additional step required to make the weighted kernel density estimator work is to establish an appropriate bandwidth for the product kernel. Here we show that the *rule-of-thumb* strategy can be efficiently applied here by leveraging the abstract kernel centers.

By *rule-of-thumb*, the lth dimension bandwidth of the product kernel is determined by the weighted standard deviation of the kernel centers on lth dimension σ^l computed by:

$$\sigma^l = \sqrt{\sum_{m=1}^{k} \omega(c_{c_m})(c_{c_m}^l - \mu^l)^2}, \tag{15}$$

where

$$\mu^l = \frac{\sum_{m=1}^{k} \omega(c_{c_m}) c_{c_m}^l}{k}, \tag{16}$$

and $c_{c_m} \in kNN(x_i, \mathbb{AKC}(S^{W_c}))$.

3.3.2 Discussion on Effectiveness and Efficiency

Effectiveness. Our aKDE builds robust density estimator based on the observation that real world datasets typically can be represented by tight data clusters because of the skewness of the data distribution. This is confirmed by our experiments using real datasets. If a dataset is uniform, then traditional sampling-based KDE tends to be effective as well.

Efficiency. As shown, the time complexity of KDE is $\mathcal{O}(nm)$, with n is number of data points and m is the number of kernel centers. Since aKDE dramatically reduces the number of kernel centers, it significantly speeds up the KDE computation. On the other hand, data clustering introduces extra computation overhead. As we will introduce in Section 3.3.5, we design a low complexity stream clustering algorithm that successfully clusters the data points by processing each point only once. This overhead is significantly outweighed by the saved KDE computation costs. Therefore, overall aKDE significantly outperforms the traditional KDE in computation costs – as shown in Section 3.4.1.

3.3.3 Outlier Detector

Our top-N local outlier detector fully leverages the data clusters produced for our aKDE. It is based on our *stable density* observation described below.

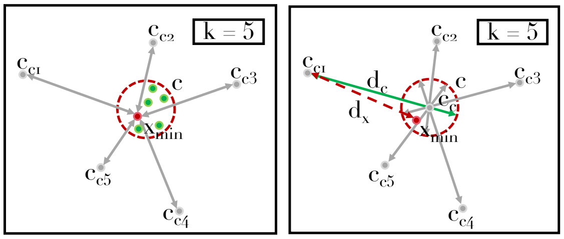
Stable Density Observation. Data points in a tight data cluster are close to each other. Therefore, they tend to share the same kernel centers and have similar probability densities. By the definition of local outliers, the outlierness score of a point x depends on the relative density at x in contrast to those at its neighbors. Therefore, these points tend to have similar outlierness scores. Since outliers only correspond to small subset of points with the largest outlierness scores, it is likely that most of the data clusters do not contain any outlier.

Assume we have a method that can approximate the largest (upper bound) and smallest (lower bound) outlierness scores for the points in each data cluster. Then by leveraging the bounds, the data clusters that have no chance to contain any outlier can be quickly identified and pruned from outlier candidate set without any further investigation. More specifically, if the upper bound outlierness score of a data cluster c_i is smaller than the lower bound outlierness score of a data cluster c_j, then the whole c_i can be pruned (under the trivial premise that c_j has no fewer than N points). This is so because there are at least N points in the dataset that have outlierness scores larger than any point in c_i.

Leveraging this observation, we now design an efficient local outlier detection strategy. The overall process is given in Algorithm 1. We first rank and then prune data clusters based on their upper and lower KLOME score bounds. As shown, a small KLOME score indicates large outlier possibility. Therefore, the upper KLOME bound corresponds to the lower outlierness score bound. Similarly, the lower KLOME bound corresponds to the upper outlierness score bound. Therefore, if the lower KLOME bound of a cluster c_i is higher than the upper KLOME bound of another cluster c_j, all points in c_i can be pruned immediately. Only the clusters with a small lower KLOME bound (large outlierness score upper bound) are subject to further investigation. The densities and KLOME scores at the data point-level are computed only for the data points in these remaining clusters. Finally, the top-N results are selected among these points by maintaining their KLOME scores in a priority queue.

3.3.4 Bounding the KLOME Scores

Next, we present an efficient strategy to establish the upper and lower KLOME bounds for each given data cluster.



(a) Lowest KLOME in cluster c (b) Lower bound of cluster c

Figure 18: An example of lower KLOME bound.

By definition, the KLOME score of a point x_i corresponds to *z-score*($\tilde{f}(x_i)$, X), where X refers to the densities at x_i's kernel centers. Since the points in the same cluster c_i typically share the same kernel centers, the data point $x_{min} \in c_i$ with the minimal density determines the lower bound KLOME score of the entire cluster c_i. Similarly the upper bound is determined by the point x_{max} with the maximal density. Obviously it is not practical to figure out the lower/upper bound by computing the densities at all points and thereafter finding x_{min} and x_{max}.

Lower bound. We now show that by utilizing the statistical property of each data cluster – more specifically the *radius*, the bounds can be derived in constant time. Here we use the lower bound as example to demonstrate our solution.

Lemma 5. *Given a data cluster c_i, its k nearest kernel centers $\{c_{c_1}, \cdots, c_{c_k}\}$ and the data point x_{min} which has the minimum density among all points in c_i, $\tilde{f}_{min}(c_i) \leq \tilde{f}(x_{min})$, where $\tilde{f}_{min}(c_i)$ = $\sum_{j=1}^{k} \omega(c_{c_j}) K_h(|c_{c_i} - c_{c_j}| + r)$. Here r is the radius of c_i and c_{c_i} is the centroid of c_i.*

Proof. The density contribution $K_h(|x_i - c_{c_j}|)$ is inversely proportional to the distance between the evaluated point x_i and the kernel center c_{c_j}. The longer the distance, the smaller the density contribution is from the kernel center. The radius r of a cluster c_i is the distance from c_i's centroid c_{c_i} to the furthest possible points in c_i. The longest possible distance from a kernel center c_{c_j} to any point in c_i is denoted as d_c = $|c_{c_i} - c_{c_j}| + r$. The distance from c_{c_1} to x_{min} is denoted as d_x = $|c_{c_j} - x_{min}|$. $d_c \geq d_x$ by the triangle inequality. Therefore $K_h(d_x) \leq K_h(d_c)$. This holds for any kernel center c_{c_j}. Therefore $\tilde{f}_{min}(c_i)$ = $\sum_{j=1}^{k} \omega(c_{c_j}) K_h(|c_{c_j} - c_{c_i}| + r) \leq \tilde{f}(x_{min})$. □

Intuitively, the density at a data point is measured by the summation of the density contributions of all relevant kernel centers. The summation of the density contribution from each kernel center c_{c_j} to the point x_j that is the point furthest to c_{c_j} in c_i is guaranteed to be smaller or equal to the density at point x_{min}. This is so because the distance from x_{min} to each kernel center c_{c_j} cannot be larger than the distance between c_{c_j} and x_j.

According to Lemma 5, given the radius of a data cluster c_i and its k nearest kernel centers $c_{c_1} \cdots c_{c_k}$, the **lower KLOME bound** of cluster c_i is computed as:

$$KLOME_{low}(c_i) = z\text{-}score(\tilde{f}_{min}(c_i), \{\tilde{f}(c_{c_1}) \cdots \tilde{f}(c_{c_k})\}), \tag{17}$$

Upper Bound. Similarly, we can show that the maximal local density at a cluster c_i, denoted by $\tilde{f}_{max}(c_i)$, can be obtained based on the shortest distance from each kernel center to the points in c_i.

$$\tilde{f}_{max}(c_i) = \sum_{j=1}^{k} \omega(c_{c_j}) K_h(|c_{c_j} - c_{c_i}| - r) \tag{18}$$

Accordingly, the upper KLOME bound of each cluster c_i $KLOME_{up}(c_i)$ is derived based on $\tilde{f}_{max}(c_i)$.

$$KLOME_{up}(c_i) = z\text{-}score(\tilde{f}_{max}(c_i), \{\tilde{f}(c_{c_1}) \cdots \tilde{f}(c_{c_k})\}), \tag{19}$$

3.3.5 The Stream Data Extractor

The stream data extractor features a lightweight stream clustering algorithm that clusters the similar data points together. As the clusters are continuously constructed and incrementally maintained, the statistics needed by both aKDE and inlier pruning, namely the *cardinality*, the *centroid*, and the *radius* of the cluster, must also be continuously generated. We thus refer to this as dual-purpose clustering.

The dual-purpose clustering is based on two key ideas: *additive meta data* and *pane-based meta data maintenance*.

The **additive meta data** is inspired by Micro-cluster [4] – a popular stream clustering approach. The idea is that by maintaining meta data that satisfies the additive properties, the statistics required by both the density estimator and the outlier detector can be computed in constant time whenever the window evolves.

Definition 15. *A **cluster** c_i in a d-dimensional data set $S_{W_c} = \{x_1, \cdots, x_m\}$ corresponding to the data in the current window W_c of stream S is represented as a 4-tuple set $\{M, LS, R_{min}, R_{max}\}$ where M denotes the cardinality of the cluster, $LS =< \sum_{i=1}^{m} x_i^1, \cdots, \sum_{i=1}^{m} x_i^d >$ is the linear sum of the points by dimension, $R_{min} =< x_{min}^1, \cdots, x_{min}^d >$ and $R_{max} =< x_{max}^1, \cdots, x_{max}^d >$ are the minimum and maximum values of the points in each dimension.*

Cardinality and Centroids for aKDE. In Definition 15, M refers to data cardinality of cluster c_i. M is used to compute the *weight* (Equation 14) and the *centroid* of the abstract kernel center. The linear sum LS is used to compute the *centroid* of cluster $c_i = \frac{LS}{M}$.

The Radius for Inlier Pruning. R_{min} and R_{max} representing the minimal and maximal values in each dimension are utilized to compute the radius of cluster c_i. Radius is a key statistic needed by our outlier detector to quickly prune the clusters from the outlier candidate set.

Since the radius is defined as the distance from the centroid c_{c_i} to its furthest point in cluster c_i, the radius changes whenever the centroid changes. All points in c_i then have to be re-scanned to find the point "furthest" from the new centroid. This, being computational expensive, is not acceptable in online applications.

The remedy comes from our carefully selected product kernel function. In the product kernel, each dimension has its own customized bandwidth. Accordingly, we only need the radius on each single dimension to estimate the bandwidth instead of the radius over the multi-dimensional data space. We now show that updating the radius in each one-dimensional space can be accomplished in constant time by utilizing R_{min} and R_{max}.

Lemma 6. *Given a new centroid c_{c_i} and its value on lth dimension v^l, the radius of c_i on the lth dimension $r^l = max\{\mid v^l - x_{min}^l \mid, \mid x_{max}^l - v^l \mid\}$, where $x_{min}^l \in R_{min}$ and $x_{max}^l \in R_{max}$.*

Pane-based meta data maintenance. The pane-based meta data maintenance strategy [64] is utilized to effectively update the meta data for each cluster as the window slides. Given the window size $S.win$ and slide size $S.slide$, a window can be divided into $\frac{S.win}{gcd(S.win, S.slide)}$ small panes where gcd refers to greatest common divisor. The meta data of a cluster c_i is maintained at the pane granularity instead of maintaining one meta data structure for the whole window. Since the data points in the same pane arrive and expire at the same slide pace, the meta data can be quickly computed by aggregating the meta data structures maintained for the unexpired panes as the window moves. This process is illustrated in Figure 19. Since the meta data satisfies the additive property, the computation can be done in constant time. In this way, no explicit operation is required to handle the expiration of outdated data from the current window. Therefore, our stream clustering algorithm only needs to exclusively deal with the new arrivals.

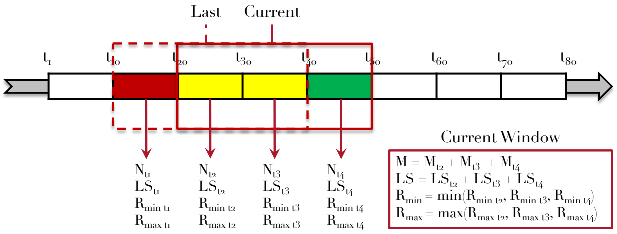
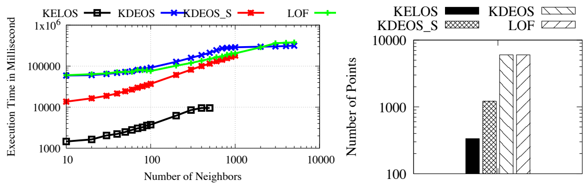

Last
Current
t1
t10
t20
t30
t40
t50
t60
t70
t80
Current Window
N_{t1} LS_{t1} $R_{min\ t1}$ $R_{max\ t1}$
N_{t2} LS_{t2} $R_{min\ t2}$ $R_{max\ t2}$
N_{t3} LS_{t3} $R_{min\ t3}$ $R_{max\ t3}$
N_{t4} LS_{t4} $R_{min\ t4}$ $R_{max\ t4}$
$M = M_{t2} + M_{t3} + M_{t4}$
$LS = LS_{t2} + LS_{t3} + LS_{t4}$
$R_{min} = min(R_{min\ t2}, R_{min\ t3}, R_{min\ t4})$
$R_{max} = max(R_{max\ t2}, R_{max\ t3}, R_{max\ t4})$

Figure 19: An example of an evolving cluster.

The Dual-Purpose Stream Clustering Algorithm. Once a new data point x arrives, DSC first finds its nearest cluster according to the distance of x to all the centroids. If the distance from x to its nearest cluster c_i denoted as $dist(x, c_{c_i})$ is smaller than a radius threshold θ, x is inserted into c_i. The corresponding *4-tuple* meta data is updated accordingly. On the other hand, if $dist(x, c_{c_i}) > \theta$, a new cluster will be created.

3.4 Experimental Results

3.4.1 Efficiency on Real Data Streams



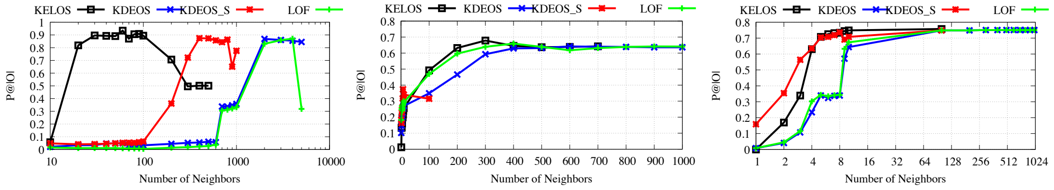
(a) Execution Time (b) Memory

Figure 20: Efficiency on HTTP.

In this section, we report the end-to-end execution time as well as the memory consumption of different methods by varying the number of neighbors k on HTTP dataset.

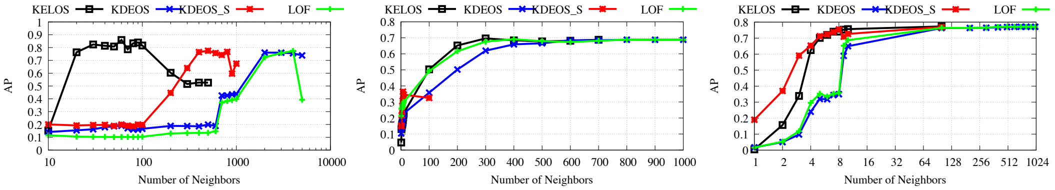
In this set of experiments, we fix the window size for HTTP as 6,000 and vary the number of neighbors k. The k parameter defines the number of neighbors to be considered in the computation of outlierness score for each point. It is critical to the effectiveness and efficiency of local outlier detection. The radius threshold θ of KELOS is set as 0.095. The sampling rates of KDEOS_S is set as 10%. This relatively high sampling rate ensures that KDEOS_S always has more than k kernel centers to use as k increases.

Execution Time. As shown in Figure 20 (a), KELOS is more than 2 orders of magnitude faster than the alternatives. The line of KELOS stops at 800, because KELOS uses our cluster-based aKDE approach. The number of the kernel centers is restricted by the number of the clusters. Similarly, the line of KDEOS_S stops at 1,000 due to the limited number



(a) HTTP (b) Yahoo! A1 (c) Yahoo! A2

Figure 21: $P@|O|$ of varied number of neighbors k. Note the maximum k that each method can reach is different. For LOF, it depends on the total number of data points. For KDE-based methods, it depends on the number of kernel centers available.



(a) HTTP (b) Yahoo! A1 (c) Yahoo! A2

Figure 22: AP of varied number of neighbors k. Note the maximum k that each method can reach is different.

of samples. Among these algorithms LOF and KDEOS are the slowest and have the similar performance, because their time complexities are both quadratic in the number of points in each window. KDEOS_S is much faster than KDEOS and LOF, because KDEOS_S only utilizes the points uniformly sampled from the data as kernel centers. Searching the k nearest kernel centers from the small sampled kernel center set is much faster than the k nearest kernel center search among all points in each window. However, KDEOS_S is still at least 1 order of magnitude slower than KELOS. This is because, in order to satisfy the accuracy requirement, the number of the sampled kernel centers has to be large enough to represent the distribution of the data stream. The aKDE approach of KELOS only uses the centroid of each cluster as abstract kernel center, while the number of the clusters tends to be much smaller than the number of the sampled kernel centers. Furthermore, KELOS effectively prunes most of the inliers without conducing the expensive density estimation, while in contrast, KDEOS_S has to compute the outlierness score for each and every data point.

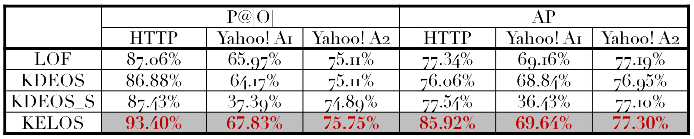
3.4.2 Effectiveness Evaluation

In this set of experiments, we compare the accuracy of the proposed method to the baselines on the labeled public datasets by varying the most important variables.

Varying Number of Neighbors k. This set of experiments is conducted on the HTTP, Yahoo! A1, and Yahoo! A2 datasets. The radius thresholds of KELOS for HTTP, Yahoo!

A1, and Yahoo! A2 are set as 0.095, 0.1 and 40. The window sizes of HTTP, Yahoo! A1, and Yahoo! A2 are set as 6,000, 1,415, and 1,412 respectively. The sampling rates of KDEOS_S is set as 10% for the similar reason with the efficiency evaluation (Section 3.4.1).

Table 1: Accuracies on labeled real datasets.



	P@|O|			AP		
	HTTP	Yahoo! A1	Yahoo! A2	HTTP	Yahoo! A1	Yahoo! A2
LOF	87.06%	65.97%	75.11%	77.34%	69.16%	77.19%
KDEOS	86.88%	64.17%	75.11%	76.06%	68.84%	76.95%
KDEOS_S	87.43%	37.39%	74.89%	77.54%	36.43%	77.10%
KELOS	**93.40%**	**67.83%**	**75.75%**	**85.92%**	**69.64%**	**77.30%**

Table 1 shows the peak $P@|O|$ and AP for each approach on each dataset. KELOS outperforms all other approaches in all cases.

Figures 21 and 22 further demonstrate the trend of $P@|O|$ and AP as k varies. Figure 21(a) shows the results on the HTTP dataset. For our KELOS, as k increases, the $P@|O|$ increases until k reaches 80. It then starts decreasing after k is larger than 100. This confirms our observation that using as many as possible kernel centers in the density estimator does not always lead to more accurate density estimation. This justifies our decision of adopting the balloon kernel that only takes the close kernels into consideration when estimating the density at a point p. Overall KDEOS, KDEOS_S, and LOF show the similar trend. Compared to KELOS they have to use a much larger k to get relative high accuracy. Interestingly, KDEOS_S has similar $P@|O|$ with KDEOS. This shows that sampling-based KDE works well on large datasets.

The trends on the Yahoo! A1 and A2 datasets are different from that on the HTTP dataset as shown in Figures 21 (b) and 21 (c). Similar to the HTTP dataset, the $P@|O|$ continuously increases and gets stable after k reaches certain value. Furthermore, we observe that KDEOS_S works poor on Yahoo! A1 dataset. The reason is that the Yahoo! A1 and A2 datasets are relatively small. The samples drawn from small dataset often are not sufficient to represent the distribution of the whole dataset.

The trends of AP are similar to the trends of $P@|O|$ on all datasets as shown in Figure 22. Overall, KELOS is as accurate or more accurate than alternative approaches. Furthermore, compared to the alternatives, KELOS uses a smaller k to achieve high accuracy. This also contributes to the performance gain of KELOS in execution time.

3.5 Related Works

Local Outlier Factor. Local outlier detection has been extensively studied in the literature since the introduction of the Local Outlier Factor (LOF) semantics [18]. A detailed survey of LOF and its variations can be found in [21]. The concept of local outlier, LOF in particular, has been successfully applied in many applications [21]. However, LOF requires kNN search for each and every data point and needs multiple iterations over the entire dataset to compute these LOF values. For this reason, to support continuously evolving streaming data, [87] studied how to quickly find the points whose LOF scores are influenced by new arrivals or expired data to avoid re-computing the LOF score for each point as the window slides. However as the velocity of the stream increases, most of the points in a window will be influenced. Therefore this approach does not scale to high volume streaming data. In [99] an approximation approach was designed to support LOF in streaming data that focuses on the memory efficiency. However, the more important problem in stream mining, namely the CPU efficiency, was overlooked, which now is instead the focus of our work.

Efficient Kernel Density Estimation. Kernel density estimation is considered as a quadratic process $\mathcal{O}(nm)$ with n the total number of data points and m the number of kernel centers in the probability density function. Previous efforts have been made to accelerate this process while still providing accurate estimation, such as utilizing sampling [102]. [122, 44] designed a method that incrementally maintains a small and fixed size of kernel centers to perform density estimation over data streams. However, to ensure the accuracy of density estimation over skewed dataset, the sample size has to be large. Therefore it cannot solve the efficiency problem of KDE in our context.

[37] studied the density-based classification problem. It proposed a pruning method that correctly classifies the data without estimating the density for each point by utilizing a user-defined density threshold. However, this pruning method can not be applied to solve our problem, since a point with low density is not necessarily an outlier based on the local outlier semantics we target on.

Outlier Detection using KDE. For each point in the current window of a sliding window stream, [107] utilizes KDE to approximate the number of its neighbors within a certain range. This information is then utilized to support distance-based outlier detection and LOCI [82]. It directly applies off-the-shelf KDE method on each window. No optimization technique is proposed to speed up KDE in the streaming context.

[59] is the first work that studied how to utilize KDE to detect local outliers in static dataset. This later was improved by [100] to be better aligned with LOF semantic. Each data point's density is estimated based upon the surrounding kernel centers only, therefore called local density. Instead of considering outliers only based on their density value, data points are measured based on the density in contrast to their neighbors. However, this work does not focus on improving the efficiency of KDE. Nor it considers streaming data. As confirmed in our experiments, it is orders of magnitude slower than our KELOS.

Other Streaming Outlier Detection Approaches. LEAP [23] and Macrobase [8] scale distance-based and statistical-based outlier detection methods respectively to data streams where they rely on either the number of neighbors in a certain distance range or the frequency of each data point to detect outliers. More specifically in these works, a data point is considered to be an outlier if its neighbor count (or frequency) is lower than a *global* cut-off threshold. However, applying such a *global* cut-off threshold uniformly to the whole dataset is not ineffective in handling skewed datasets [33]. For example, a point with a small number of neighbors is not necessarily an outlier if it is located in a relative sparse subspace of the dataset. On the other hand, a point with a relative large number of neighbors might instead be an outlier, if it is located in a dense subspace and other points have many more neighbors than it.

3.5.1 Conclusion

We present the first solution called KELOS for continuously monitoring top-N KDE-based local outliers over sliding window streams. First, we propose the KLOME semantics, effective in measuring the outlierness scores of streaming data. Furthermore, continuous detection strategy is devised that efficiently supports the KLOME semantics by leveraging the key properties of KDE. Using both real world and synthetic datasets we demonstrate that KELOS is 2-3 orders of magnitude faster than the baselines, while being highly effective in detecting outliers from data stream.

4 Text Modeling and Generation

Manuscript

1. **Xiao Qin**, Cao Xiao, Tengfei Ma, Tabassum Kakar, Susmitha Wunnava, Xiangnan Kong, Elke Rndensteiner and Fei Wang. *Integrating Neural Language Model with Supervised Topic Modeling: Application to Clinical Narrative Modeling*. (In submission.)

2. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Cansu Sen, Elke A. Rundensteiner and Xiangnan Kong. *Adverse Drug Event Detection from Electronic Health Records Using Hierarchical Recurrent Neural Networks with Dual-Level Embeddings*. **Drug Safety**'19, 1-10.

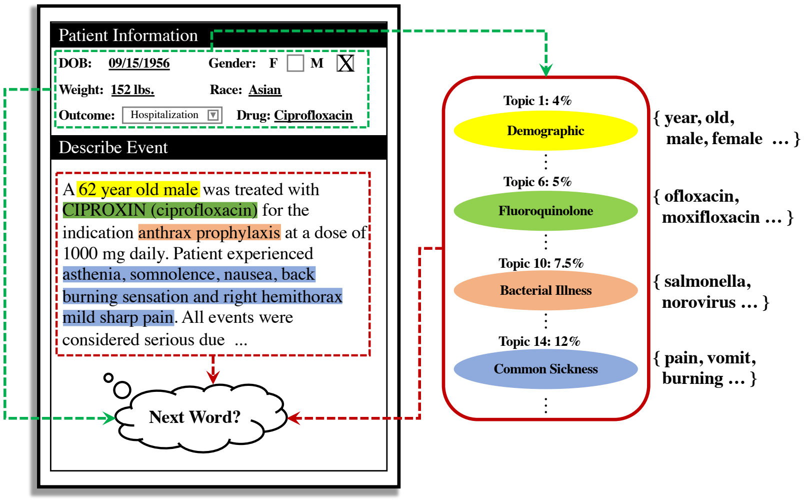
3. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Elke A. Rundensteiner and Xiangnan Kong. *Deep Learning Strategies for Automatic Detection of Medication and Adverse Drug Events from Electronic Health Records.* **AMIA**'18.

4. Susmitha Wunnava, **Xiao Qin**, Tabassum Kakar, Elke A. Rundensteiner and Xiangnan Kong. *Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records*. **MADE**'18, 48-56.

4.1 Introduction

This decade has seen an explosion in the amount of digital information presented in the electronic health records (EHR), in part due to the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 which promotes the adoption and meaningful use of health information technology [12]. In 2015, 84% of hospitals in the United States adopted at least a *Basic EHR system* which represents a 9-fold increase since 2008 [45]. An EHR is a patient-centered record consisting of heterogeneous data elements, including patient demographic information, diagnoses, laboratory test results, medication prescriptions, medical images and free-form clinical narratives [118]. In particular, the clinical narratives provide a diagram that concatenates complex medical events via natural language which encode critical insight very often not presented or missed from the structured fields, e.g. description of *Challenge-Dechallenge-Rechallenge* (CDR) [104] phenomenon that verifies adverse drug reactions. The problem of text mining clinical narratives through natural language processing (NLP) has attracted increasing attention in recent years.

Language models (LMs) whose goal is to learn the joint probability function of sequences of words in a language are one of the key enablers to many NLP applications including machine translation, named entity recognition and text summarization. The capability of capturing long term relationships among text is crucial to the performance



Patient Information

DOB: 09/15/1956 Gender: F □ M ☒

Weight: 152 lbs. Race: Asian

Outcome: Hospitalization Drug: Ciprofloxacin

Describe Event

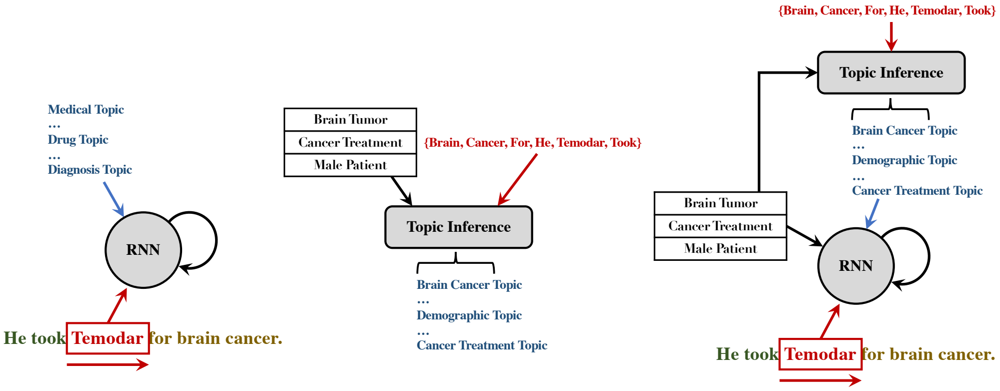
A 62 year old male was treated with CIPROXIN (ciprofloxacin) for the indication anthrax prophylaxis at a dose of 1000 mg daily. Patient experienced asthenia, somnolence, nausea, back burning sensation and right hemithorax mild sharp pain. All events were considered serious due ...

Next Word?
Topic 1: 4%
Demographic
{ year, old, male, female ... }
Topic 6: 5%
Fluoroquinolone
{ ofloxacin, moxifloxacin ... }
Topic 10: 7.5%
Bacterial Illness
{ salmonella, norovirus ... }
Topic 14: 12%
Common Sickness
{ pain, vomit, burning ... }

Figure 23: The generative process of a clinical narrative. Green dashed box highlights document meta-information, red box the latent topics of the narrative and their associated vocabulary, while red dashed box the preceding text.

of LMs [74]. Recurrent neural network (RNN) based language models in particular have demonstrated promising results in modeling complex and long dependencies. Recently, RNN based methods have been widely used in processing medical text [69]. In theory, RNNs such as Long Short-Term Memory (LSTM) [47] and Gated Recurrent Unit (GRU) [26] can "remember" arbitrarily long span of history if provided with enough capacity. However, they do not perform well on very long sequences in practice as the gradient computation for RNNs becomes increasingly ill-behaved as the expected dependency becomes longer [84]. One way of tackling this problem is to feed succinct information that encodes the semantic structure of the document such as latent topics as context to guide the modeling process [75], as illustrated in Figure 24(a). In this vein, existing works [29, 60, 112] focus on the global context obtained from the text itself, overlooking the opportunity to exploit existing document meta-information which may provide explicit insight into the global context.

Motivating Example. Let's consider the generative process of a clinical narrative describing a patient's adverse drug events as illustrated in Figure 23. Before drafting the narrative, the doctor fills out the structured template form with the "central ingredients" of the narrative such as the patient's demographics, suspected drugs, severity, etc. With this descriptive information and the observed events such as "experiencing nausea after taking Ciproxin" in mind, the doctor then composes the overall story by considering the relevant topics and their corresponding vocabulary. Finally, the narrative summarizing



(a) Contextual RNN (b) Supervised Topic Model (c) MeTRNN

Figure 24: Intuitions of different text modeling approaches.

all information is composed with appropriate words in order. This motivating example highlights the following insights: (1) Latent topic information such "Bacterial Illness" topic and its proportion in the text as the global context to guide and regulate the language modeling process; (2) Document meta-information could be leveraged to learn more accurate and relevant topic information with respect to the key medical information.

Limitations of State-of-the-Art. Contextual RNNs (cRNNs) [75] obtain topic information from latent Dirichlet allocation (LDA) [16] and feed it into an additional *feature layer* connected to the recurrent unit to guide the modeling process. To ensure that the learned topics are in favor of those that indeed improve the language modeling performance, TopicRNN [29] further extends cRNN by combing topic model and cRNN into a unified model that trains the two components simultaneously. However, these models only focus on the semantic structure inferred from the text itself. Hence, they miss the opportunity of obtaining a more complete context that also incorporates document meta-information. On this front, supervised topic models (sTMs) [96, 77, 71, 93, 94, 78, 49], illustrated in Figure 24(b), use observable document meta-information to supervise the learning of better topic representations. However, these models are bag-of-word models that do not account for word ordering, which is essential to our problem.

Challenges. To integrate the strength of sTMs into cRNNs for better clinical narrative modeling performance, the following research challenges need to be tackled: (1) *Flexible supervised topic model component.* Existing latent Dirichlet allocation (LDA) variations [96, 77, 71, 93, 94, 49] that incorporate document meta-information focus on specially constructed models. Even small changes to these ad-hoc solutions require deriving new inference methods which can be onerous for practitioners to freely experiment with different modeling assumptions. Moreover, existing solutions cannot accommodate combinations of modalities of data beyond their original intention. The lack of capability to manage arbitrary meta-data limits their effectiveness on complex inputs such as EHR which contains meta-information coded in various format. (2) *End-to-end Framework.* sTMs learn the the topics from the bag-of-words representation of the text and their corresponding meta-information. Although the learned topics representing the underlying semantic structure of a document can encode long-range dependencies for cRNNs, such topics do not reflect on information indicated by the ordering of the words (e.g "eat to live" vs. "live to eat") missing the opportunity to capture the true semantics of the text. In order to better facilitate cRNNs on sequence modeling task, establishing direct connection between sTMs to the goal of language model becomes critically important.

Contribution. To tackle the above challenges, we propose a neural language model called MeTRNN (Figure 24(c)) which enhances RNN-based language models' capability of establishing long-range dependencies by leveraging arbitrary document meta-information through their *implicit* influence via supervised latent topics and through *explicit* influence via a feature layer that directly connects to the RNN cells. It is worthwhile to highlight the following contributions of the proposed approach.

1. MeTRNN defines and explicitly models the text generative process based on the observation of the composition of the clinical narrative in an EHR.

2. MeTRNN captures the latent topics in text by leveraging the associated meta-information, which serves as the global context of the text that leads to better language modeling performance. To cope with various structured information in the EHRs, we propose a flexible supervised topic model component that can take on arbitrary meta-information.

3. We design a joint model that connects sTMs to cRNNs with an end-to-end autoencoding variational Bayes inference method using the conditional variational autoencoder framework [103]. It is a "black box" method that can be easily adjusted or extended.

4. We demonstrate the effectiveness of MeTRNN in word prediction using publicly available text datasets as well as real world Electronic Health Records (EHRs). MeTRNN achieves improvement in perplexity from 5% to 40% against baselines. We also conduct a case study that demonstrates MeTRNN's ability to learn useful

global context for better language modeling performance and more relevant topics to the structured meta-information.

4.2 Preliminary

4.2.1 RNN-based Language Models.

Traditional n-gram and feed-forward neural network-based language models make the Markov assumption about the dependencies between consecutive words where the chain rule limits conditioning to a fixed size context window. RNN-based language models overcome the Markov assumption by defining the conditional probability of each word w_t given all the previous words $w_{1:t-1}$ through a hidden state h_t:

$$\begin{aligned} p(w_t|w_{1:t-1}) &\triangleq p(w_t|h_t), \\ h_t &= f(h_{t-1}, w_{t-1}). \end{aligned} \tag{20}$$

The function $f(\cdot)$ can be a standard RNN cell or a more complex cell such as GRU or LSTM. While in principle RNN is good at remembering the long-term dependencies, in practice, training a large-scale neural network on long histories can be difficult. Contextual RNN (cRNN) [75] tackles this problem by adding a *feature layer* that regulates the model by introducing the side information as additional context. Side information refers to information in or reasoned from the text such as document topic information obtained from latent Dirichlet allocation (LDA) [16]:

$$\begin{aligned} p(w_t|w_{1:t-1}) &\triangleq p(w_t|h_t, x), \\ h_t &= f(h_{t-1}, w_{t-1}, x), \end{aligned} \tag{21}$$

where x denotes the side information.

4.2.2 Latent Dirichlet Allocation.

Probabilistic topic models are a family of models that aim to find groups of words that tend to co-occur within a document. These groups of words are called topics. Each topic β_k represents a probability distribution that puts most of its mass on this topic related vocabulary. A document can then be represented as a mixture over these topics $\beta = (\beta_1 \cdots \beta_K)$. β is said to encode the global semantics. Topic models are *bag-of-words* models where the word order is ignored.

For the most popular topic model, latent Dirichlet allocation (LDA) [16], its generative process of a document $w_{1:T}$ is:
The marginal likelihood of a document $w_{1:T}$ is:

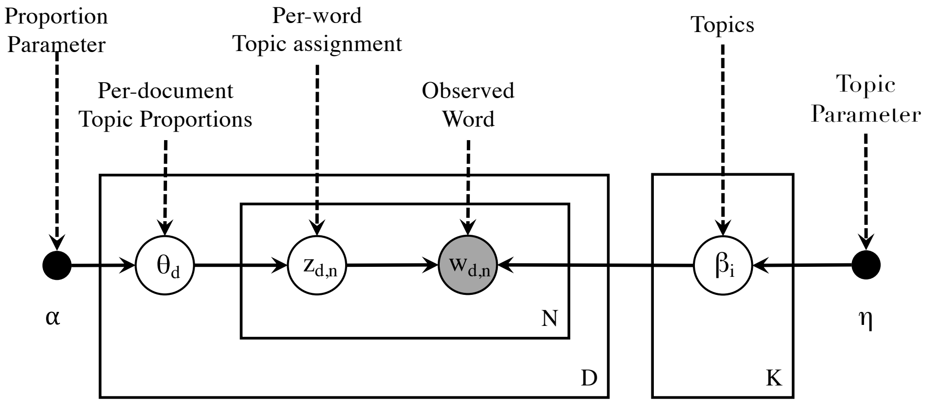


Figure 25: Plate notation for LDA with Dirichlet-distributed topic-word distributions. D denotes the number of documents in a corpus, N is the number of words in a document and K is the specified number of topics.

```
for each document $w_{1:T}$ do
    Draw topic distribution $\theta \sim$ Dirichlet($\alpha$)
    for each word $w_t$ do
        Draw topic assignment $z_t \sim$ Multinomial(1,$\theta$)
        Draw word $w_t \sim$ Multinomial(1,$\beta_{z_t}$)
    end
end
```

$$p(w_{1:T}|\alpha, \beta) = \int_{\theta} \Big(\prod_{t=1}^{T} \sum_{z_t=1}^{K} p(w_t|z_t, \beta) \Big) p(z_t|\theta) d\theta. \tag{22}$$

Posterior inference over the hidden variables θ and z is intractable due to the coupling between θ and β under the multinomial assumption. A popular approximation for efficient inference is *mean field variational inference* [15] which sidesteps this issue by introducing free variational parameters γ over θ and ϕ over z and dropping the edge between them. This results in an approximate variational posterior $q(\theta, z|\gamma, \phi) = q_{\gamma}(\theta) \prod_t q_{\phi}(z_t)$, which is optimized to best approximate the true posterior $p(\theta, z|w_{1:T}, \alpha, \beta)$. The optimization problem is to minimize the *evidence lower bound* (ELBO):

$$\begin{aligned} \mathcal{L}(\gamma, \phi|\alpha, \beta) = -D_{KL}[q(\theta, z|\gamma, \phi)||p(\theta, z|\alpha)] + \\ \mathbb{E}_{q(\theta, z|\gamma, \phi)}[\log p(w_{1:T}|z, \theta, \alpha, \beta)]. \end{aligned} \tag{23}$$

The first term in Equation 23 tries to match the variational posterior over latent variables to the prior on the latent variables, while the second term ensures that the variational posterior favors values of the latent variable that are good at explaining the data. Recently, several methods are proposed to "black box" the inference by using the variational autoencoder framework [56]. The variational parameters are computed by using a neural network called an inference network that takes the observed data as input. The second term in Equation 23 is referred to as a *reconstruction term* in the autoencoder network. The expectation w.r.t. q is computed by using a Monte Carlo estimator, called *reparameterization trick*.

Supervised topic models (sTMs) [13] are a group of topic models for incorporating side information. They can be categorized into two classes, namely, *downstream supervised topic* (DsTM) and *upstream supervised topic model* (UsTM). In a DsTM such as [34, 113, 114, 79, 14], meta-information, a.k.a. the response, is predicted based on the latent representation of the document, whereas in a UsTM such as [96, 72, 77, 78, 30] the response variable is being conditioned on to generate the latent representation.

4.3 The Proposed Approach

Next, we describe our proposed supervised topic compositional neural language model (MeTRNN). The realization of MeTRNN is a deep learning framework that integrates a sTM like component into a cRNN for improving the language modeling capacity. First, we introduce the general principle of how we utilize the meta-information in our model. Second, we formally define the MeTRNN model. Third, we propose an inference method for MeTRNN. Finally, we discuss our strategy for training MeTRNN.

4.3.1 Document Meta-Information.

Document meta-information, as motivated in the clinical narrative scenario, provides the central ingredients of the narrative text as well as a clue in semantic structure of the entire narrative. Based on this observation, we design our model such that meta-information has both *explicit* and *implicit* influence on language modeling. For *explicit* influence, we add a *feature layer* similar to [75] that takes meta-information directly connected to the recurrent unit in RNN. For *implicit* influence, we introduce a sTM like component where the meta-information is used as a response to produce relevant topic information. In this study, we adopt the idea of UsTM approach where meta-information is being conditioned on to generate the topic information of the narrative. The widely used UsTM approach is considered closer to the the generative process [78] in the clinical narrative scenario where all meta-information is pre-defined and is used for defining the topics. MeTRNN works with arbitrary meta-information as long as there exists a vector representation of such information. The exact computation of *explicit* and *implicit* influence is formalized next.

4.3.2 MeTRNN Model.

We define MeTRNN as a generative probabilistic model of an EHR corpus. The idea is that the semantic structure of a document is represented as a random mixture of latent topics conditioned on some document meta-information. Each topic is characterized by a distribution over words. The distribution of a word in the text narrative is then estimated given all the preceding words, latent topics and the document meta-information. For each document $d = (x_d, w_{1:T})$ where x_d is a vector that encodes the meta-information of d, e.g. representation of the structured information in an EHR, and $w_{1:T}$ is the associated narrative text, the generation process of $w_{1:T}$ is defined as follows:

```
for each document $d = (x_d, w_{1:T})$ do
    I. Draw a topic proportion vector $\theta \sim p(\theta|x_d)$
    for each word $w_t$ do
        II. Compute the hidden state $h_t = f(w_{t-1}, h_{t-1})$
        III. Draw word $w_t \sim p(w_t|h_t, \theta, x_d)$ where $p(w_t{=}i|h_t, \theta, x_d) \propto \exp(v_i^\top h_t{+}b_i^\top \theta{+}c_i^\top x_d)$
    end
end
```

θ is drawn from a Dirichlet distribution over θ conditioned on the document meta-information x_d. θ is the topic proportions influenced by the document meta-information which encodes the semantic structure of the document d. f computes the hidden state of the RNN (Equation 20) based on the previous word and hidden state. The current hidden state h_t encodes the local dynamics of the composed word sequence up to time $t-1$. Finally, the next word w_t is decided based on the hidden state h_t, topic proportions θ and document meta-information x_d through an additive procedure. In [75], x_d and θ are referred as additional side information to affect the word choices in the language model. Following [29], instead of passing them into the hidden state of the RNN, they are used as bias to have their global semantic contributions to the word choices clearly separated from those of local dynamics. The contextual contribution is measured by the summation of the dot products between θ, x_d and respective latent word vectors $b_i \in W_{\theta w}$ and $c_i \in W_{mw}$ for the ith vocabulary word.

The unrolled graphical representation of MeTRNN is depicted in Figure 26. The log marginal likelihood of the word sequence $w_{1:T}$ composing a document d is:

$$\log p(w_{1:T}|x_d) = \log \int p(\theta|x_d) \prod_{t=1}^{T} p(w_t|h_t, \theta, x_d) \mathrm{d}\theta \tag{24}$$

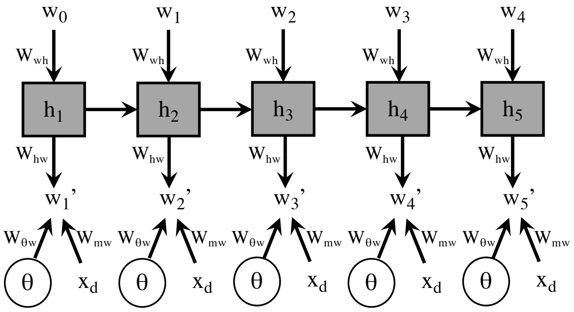


Figure 26: The unrolled MeTRNN architecture: $w_0, \cdots, w_4$ are words in the document, h_t is the state of the RNN at time step t, θ is the latent representation of the EHR and x_d is the meta-information.

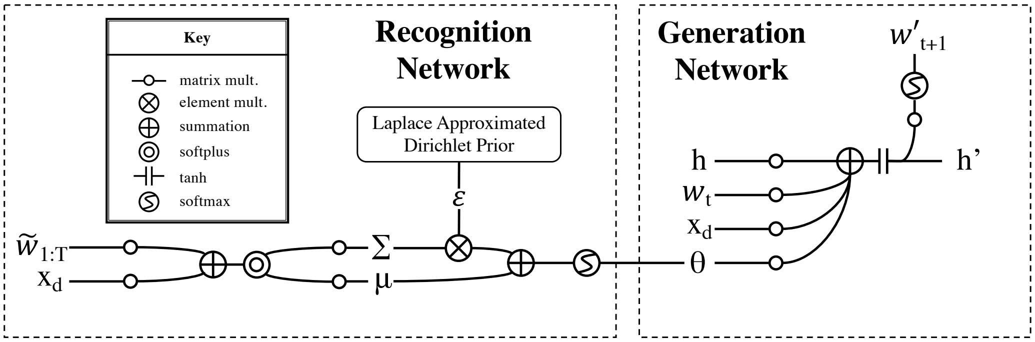


Figure 27: An example of MeTRNN inference network with a vanilla recurrent neural network cell. The input of the recognition network are $\tilde{w}_{1:T}$ (or $\tilde{w}_{1:t}$) the bag-of-words representation of the text and x_d the vector representation of the meta-information. The input of the generation network at time t includes the hidden state h from the previous time stamp, current word w_t, topic vector θ and meta-information x_d.

4.4 The Model Inference.

Since directly optimizing Equation 24 is intractable, we use variational inference for approximating this marginal. Let $q(\theta)$ be the variational distribution on the marginalized θ. The variational lower bound of the model is written as follows:

$$\begin{aligned} \log p(w_{1:T}|x_d) \geq -D_{KL}(q(\theta|\tilde{w}_{1:T}, x_d)||p(\theta|x_d))+ \\ \mathbb{E}_{q(\theta|w_{1:T},x_d)}[\log p(w_{1:T}|\theta, x_d)]. \end{aligned} \tag{25}$$

ELBO is written as:

$$\begin{aligned} \mathcal{L}(x_d, w_{1:T}) \triangleq -D_{KL}(q(\theta|\tilde{w}_{1:T}, x_d)||p(\theta|x_d))+ \\ \mathbb{E}_{q(\theta|w_{1:T},x_d)}\Big[\sum_{t=1}^{T}\log p(w_t|h_t, \theta, x_d)\Big]. \end{aligned} \tag{26}$$

Following the proposed conditional variational autoencoder (CVAE) [103], we choose the form of $q(\theta)$ to be a "black box" inference network using a feed-forward neural network. Specifically, the MeTRNN inference network consists of a recognition network $q(\theta|\tilde{w}_{1:T}, x_d)$ where $\tilde{w}_{1:T} \in d$ is a bag-of-words representation of $w_{1:T}$, a prior network $p(\theta|x_d)$ and a generation network $p(w_{1:T}|\theta, x_d)$ that reconstructs the word sequence. In our formulation, the prior of the latent variable θ is modulated by the meta-information. This can be relaxed to make the latent variables statistically independent of x_d [55], i.e., $p(\theta|x_d) = p(\theta)$. We show the graphical representation of MeTRNN inference network in Figure 27.

$q(\theta)$ is reparameterized with a deterministic, differentiable function $g(\cdot, \cdot, \cdot)$, whose arguments are meta-information x_d, words $\tilde{w}_{1:T}$ and the noise variable ϵ. This, known as *reparameterization trick* [56], allows for error backpropagation through the latent variables, essential in variational autoencoder training. In MeTRNN, the latent variable θ follows a Dirichlet distribution as suggested by the classical topic models [16] due to its flexibility. However, Dirichlet distribution does not belong to the *location-scale* family which makes *reparameterization trick* difficult to use. We solve this by constructing a Laplace approximation to the Dirichlet prior [105]. We approximate the prior distribution with $\hat{p}(\theta|\mu_1, \Sigma_1) = \mathcal{LN}(\theta|\mu_1, \Sigma_1)$ where $\mathcal{LN}$ is a logistic normal distribution,

$$\begin{aligned} \mu_{1k} &= \log \alpha_k - \frac{1}{K} \sum_i \log \alpha_i, \\ \Sigma_{1kk} &= \frac{1}{\alpha_k}\Big(1 - \frac{2}{K}\Big) + \frac{1}{K^2} \sum_i \frac{1}{\alpha_i}, \end{aligned} \tag{27}$$

with $\alpha = (\alpha_1, \cdots, \alpha_K)$ being the parameter of the Dirichlet prior and K the dimension of the hidden space, a.k.a. specified number of topics. Finally, $\theta = g(x_d, w_{1:T}, \epsilon), \epsilon \sim \mathcal{N}(0, \mathrm{I})$.

According to the defined prior network, the input of the recognition network $\tilde{w}_{1:T}$ and the meta-information vector x_d is first projected into a K-dimensional latent space. Specifically, we have:

$$\begin{aligned} q(\theta|\tilde{w}_{1:T}, x_d) &= \mathcal{LN}(\theta|\mu(\tilde{w}_{1:T}, x_d), diag(\sigma^2(\tilde{w}_{1:T}, x_d))), \\ \mu(\tilde{w}_{1:T}, x_d) &= W_{w\mu}\tilde{g}(\tilde{w}_{1:T}) + W_{m\mu}\tilde{g}(x_d) + b_\mu, \\ \log \sigma(\tilde{w}_{1:T}, x_d) &= W_{w\sigma}\tilde{g}(\tilde{w}_{1:T}) + W_{m\sigma}\tilde{g}(x_d) + b_\sigma, \end{aligned} \tag{28}$$

where $\tilde{g}(\cdot)$ denotes the feed-forward neural network. The weight matrices $W_{w\mu}, W_{m\mu}, W_{w\sigma}, W_{m\sigma}$ and biases b_μ, b_σ are shared across documents. Each document has its own parameter setting $\mu(\tilde{w}_{1:T}, x_d)$ and $\sigma(\tilde{w}_{1:T}, x_d)$ resulting in a unique distribution $q(\theta|\tilde{w}_{1:T}, x_d)$ for each document. The output of the inference network is a topic proportion vector θ that represents the global semantics of the document.

The generation network is in the form of a recurrent neural network. It learns the local dynamics of the word sequence for each topic proportion vector θ. Here we show the specification with a vanilla RNN cell and it can be easily extended to other structures such as a GRU or LSTM cell:

$$\begin{aligned} h_t &= \sigma_h(W_{wh}w_{t-1} + W_{hh}h_{t-1} + b_h), \\ w_t &= \sigma_w(W_{hw}h_t + W_{\theta w}\theta + W_{mw}x_d + b_w), \end{aligned} \tag{29}$$

where $\sigma(\cdot)$ denotes the activation functions. The weight matrices $W_{wh}, W_{hh}, W_{hw}, W_{\theta w}, W_{mw}$ and biases b_h, b_w are shared across words. The hidden state of the recurrent unit, the topic proportion vector θ and the document meta-information x_d affect the output through an additive procedure.

During training, the parameters of the inference network and the model are jointly learned and updated via truncated backpropagation throughout time using the AdaGrad algorithm [32].

4.4.1 Training MeTRNN.

Each training instance for MeTRNN consists of (1) the meta-information, (2) the words in bag-of-words representation and (3) word sequence. Following [29], we truncate the document into shorter subsequences for RNN training. However, (1) and (2) still carry the information about the entire document for the subsequence.

Similar to [17], we find that using RNN as a decoder under the conditional variational autoencoder framework fails to produce meaningful information in θ due to the *vanishing latent variable problem*. Following [17], we apply a small weight on the D_{KL} term and gradually increase it during training. The idea of having a constrained D_{KL} cost in VAE to obtain better latent representations is studied in [46]. Specifically, we have:

$$\begin{aligned} \mathcal{L}(x_d, w_{1:T}) \triangleq -\beta D_{KL}(q(\theta|\tilde{w}_{1:T}, x_d)||p(\theta|x_d)) + \\ \mathbb{E}_{q(\theta|w_{1:T}, x_d)}\Big[\sum_{t=1}^{T} \log p(w_t|h_t, \theta, x_d)\Big], \end{aligned} \tag{30}$$

where β is a hyper-parameter that balances the latent channel capacity and independence constraints.

Word Prediction. In word prediction task, MeTRNN is given the preceding word sequence $w_{1:t-1}$ and the meta-information x_d from which MeTRNN has an estimation of $q(\theta|\tilde{w}_{1:t-1}, x_d)$. To predict the next word w_t, MeTRNN computes the probability distribution of w_t incrementally. After the predicted word w_t being sampled from the predictive distribution, MeTRNN update $q(\theta)$ by including w_t. MeTRNN is then go on to predict the next word w_{t+1}.

4.5 Experimental Evaluation

4.6 Experimental Setup & Methodology

We evaluate our proposed MeTRNN model with publicly available text datasets as well as EHRs by comparing its performance on word prediction tasks against other baselines. We also conduct a case study on EHRs that shows the effectiveness of MeTRNN for learning meaningful and useful topics. All methods are implemented in PyTorch [86] and trained on an Ubuntu server with Intel Xeon E-5 2680v2 @2.8GHz CPUs and Nvidia Tesla K40m GPUs. We have released the source code [1] of the models described in this paper.

Table 5: Size in number of words. M=million, K=thousand.

Dataset	Train	Valid	Test	Vocabulary
20NG	2M	248K	266K	10K
R52	465K	90K	77K	10K
MADE	306K	53K	53K	11K

4.6.1 Datasets.

For reproducibility, we use two well known labeled datasets, namely *20 Newsgroups* (20NG) [58] and *Reuters-21578* (R52) [63] for word prediction task. The category information of each document is used as the document meta-information. We also use a labeled EHR dataset *MADE* for an adverse drug event detection competition [2]. *MADE* consists of total of 1089 de-identified EHR narratives from 21 cancer patients. Each EHR comes with annotations such as medication name, adverse events, indications and other signs and symptoms. Basic statistics of the datasets are summarized in Table 5. We partition each document into tumbling windows with length of 50. 20NG and R52 datasets are preprocessed with *stopword removal* and *stemming*. MADE corpus is preprocessed with *stopword removal*.

4.6.2 Baselines.

For word prediction tasks, we compare our MeTRNN with GRU and LSTM cells denoted as MeTGRU and MeTLSTM respectively against:

• **RNNs.** LSTM and GRU, commonly used in language modeling, are proved to be superior than vanilla RNN for long documents. Therefore, we include these two as baselines.

[1] [undisclosed for review policy, repository is not visible.]

[2] `http://bio-nlp.org/index.php/announcements/39-nlp-challenges`

Table 6: Test perplexities of different models by varying the number of neurons. The lower the perplexity the better the performance. $(\cdots)$ after each perplexity indicates the ranking of the method w.r.t. the specific setting. "T" denotes the topic feature obtained from ProdLDA trained separately using AVITM and "F" denotes the document meta-information. † or ⋆ indicates that the baseline is implemented by others or ourselves.

	20 NG			**R52**			**MADE**		
Methods	**n=128**	**n=256**	**n=512**	**n=64**	**n=128**	**n=256**	**n=16**	**n=32**	**n=64**
GRU†	360.76(12)	352.79(12)	345.68(12)	163.04(15)	151.70(15)	149.20(15)	174.17(12)	122.57(12)	99.42(12)
LSTM†	352.15(11)	337.15(11)	333.95(11)	154.26(14)	145.62(14)	143.29(14)	170.81(11)	115.97(11)	98.28(11)
cRNN(T)⋆	370.26(14)	365.47(15)	353.64(15)	146.36(13)	143.13(13)	142.56(13)	177.40(14)	130.99(13)	109.18(13)
cRNN(F)⋆	363.59(13)	362.43(13)	352.12(13)	134.57(11)	134.20(11)	132.35(11)	186.87(15)	137.06(15)	110.83(14)
cRNN(T+F)⋆	371.81(15)	364.28(14)	353.37(14)	137.22(12)	134.30(12)	133.98(12)	175.90(13)	132.46(14)	113.23(15)
cGRU(T)⋆	316.93(6)	299.99(10)	280.53(5)	118.79(6)	110.20(8)	104.74(5)	151.66(8)	108.44(4)	90.87(8)
cGRU(F)⋆	314.69(3)	297.79(8)	279.21(4)	115.38(3)	109.70(6)	106.96(8)	159.96(10)	114.58(10)	93.29(10)
cGRU(T+F)⋆	320.49(7)	298.12(9)	281.78(7)	118.34(5)	111.30(9)	105.76(6)	147.36(6)	112.25(9)	92.89(9)
cLSTM(T)⋆	322.13(9)	289.54(5)	284.58(10)	119.46(7)	109.96(7)	108.42(9)	144.89(4)	108.72(5)	88.30(3)
cLSTM(F)⋆	315.36(5)	293.77(6)	281.74(6)	117.58(4)	108.56(4)	106.50(7)	158.88(9)	111.69(8)	89.52(4)
cLSTM(T+F)⋆	321.63(8)	289.14(4)	282.89(9)	127.09(8)	116.85(10)	113.63(10)	145.79(5)	108.93(6)	90.86(7)
TopicGRU⋆	315.28(4)	296.31(7)	278.13(3)	117.32(9)	108.72(5)	103.79(3)	148.66(7)	111.32(7)	90.45(6)
TopicLSTM⋆	323.31(10)	286.30(3)	282.38(8)	121.29(10)	107.72(3)	104.02(4)	144.13(3)	108.05(3)	90.20(5)
MeTGRU⋆	**309.30**(1)	283.90(2)	273.60(2)	108.29(2)	**96.34**(1)	**90.34**(1)	**139.10**(1)	101.93(2)	82.48(2)
MeTLSTM⋆	309.98(2)	**281.59**(1)	**272.29**(1)	**107.25**(1)	98.34(2)	95.13(2)	141.05(2)	**99.84**(1)	**80.73**(1)

- **Contextual RNNs.** We implemented the contextual RNN (cRNN) from [75] and extended it using LSTM and GRU cells denoted as cLSTM and cGRU respectively. We consider three features for cRNNs: (1) topic information obtained separately from ProdLDA [105] (with an existing Pytorch implementation[3]); (2) document meta-information; (3) combination of (1) and (2). Topic information is inferred from the text.

- **TopicRNNs.** We implemented TopicRNNs with LSTM and GRU cells as they have been shown to achieve better performance than the ones with vanilla RNN cell [29]. Since stopwords are excluded from our datasets, the mechanism that explicitly models stopwords is ignored. Topic information is inferred from the text.

4.6.3 Metric.

For word prediction, we measure the word *perplexity* (PPL) typical metric for language model evaluation:

$$\mathrm{PPL} = \exp\Big(-\frac{1}{T}\sum_{t=1}^{T} \log(p(w_t|w_{1:t-1}))\Big), \tag{31}$$

where T is the length of the test document. Lower PPL indicates better prediction performance.

[3] https://github.com/hyqneuron/pytorch-avitm

4.6.4 Word Prediction on 20NG and R52.

We evaluate MeTRNN against other baselines on the word prediction task by varying the complexity of the models in the number of neurons used in each layer. We use 1 RNN layer for all methods and do not apply dropout for comparison purpose. For TopicRNNs and MeTRNNs, we use a multilayer perception with 2 hidden layers for the inference network. For comparability, we specify the number of topics for TopicRNNs and MeTRNNs to be equal to the number of categories in 20NG and R52 respectively. The validation set is used for early stopping. Hyperparameters including *learning rate*, *batch size*, α (parameter of Dirichlet prior) and β (scaling parameter for D_{KL}) are properly tuned for each method with different complexities. The specific hyperparameter settings are reported in Section **??**.

As shown in Table 6, MeTRNN consistently outperforms all other baselines. In general, the models with the capability of incorporating extra context information perform better than the ones that do not account for such information. Specifically, GRU and LSTM cannot achieve lower PPL than others with the same type of recurrent units. In the experiments with R52, cRNNs conditioned on various combinations of features achieve lower PPL than GRU and LSTM. When testing cRNNs, cGRUs and cLSTMs, we find that the document meta-information can better help the model as compared to the topic features obtained from ProdLDA. The reason is the category label in these two datasets can be seen as a better representation of the semantic structure of the document. It uniquely identifies the theme of the document and the underlying vocabulary used for the content.

As opposed to using the topic information obtained separately, TopicGRU and TopicLSTM learn the latent topics simultaneously during language modeling. Although they outperform their comparable methods cGRU(T) and cGRU(T) in a few experiments, the performances are not consistent across different settings. The closest methods to MeTRNN in context information leveraged in the model are cRNN(T+F), cGRU(T+F) and cLSTM(T+F). Interestingly, these methods which take both features by simple feature concatenation do not outperform the ones that consider only one feature. Worse yet, in some cases, their PPLs are higher than all of those which take a single feature. The reason is that the topic proportions θ obtained separately from ProdLDA and the meta-information associated with the document may not entirely "agree" on each other. In an extreme case, a topic representing some common words used in corpus may not be helpful for language modeling, worse yet, may diminish the contribution of the meta-information which encodes the central ingredients of the narrative. One naive solution is to obtain topic proportions θ from a supervised topic model so that the learned topics information balance the information from the text itself as well as the meta-information. MeTRNN extends this idea by combining a supervised topic model components with the language model to make sure that the learned semantic structure is helpful for word prediction.

4.6.5 Case Study: EHR Narrative Modeling and Generation

Next we will take a deep dive into experiments on a real EHR dataset to demonstrate how MeTRNN can learn meaningful and useful topics. Besides the structured information provided with an EHR (See Introduction), the narrative text provides a full story about the medical events of a patient. Modeling such narrative text is a fundamental task for many applications in healthcare systems [27]. We conduct a case study using MADE – a labeled EHR dataset that reports adverse drug reactions. An adverse drug reaction corresponds to an unwanted and often dangerous effect caused by the administration of a drug. MADE's labels include drug name, indication, adverse reactions, etc. In this study, we use the indication as the meta-information of the narrative. In medicine, an indication is a valid reason to use a certain medication. An indication can correspond to a certain type of medication which may trigger specific reactions commonly associated with these drugs. The indication can reveal the semantics of the narrative. We include 102 unique indications in this dataset to encode a narrative's meta-information vector.

For the word prediction task on this dataset, words are not stemmed in order to generate interpretable topics. Comparing to the meta-information used for 20NG and R52, indication can capture partial or different semantics from the topic information learned from the narrative itself as confirmed by the results shown in Table 6. The cRNNs conditioned on topic feature achieves lower PPL than those conditioned on the indication feature. However, it remains true that cRNNs is not further improved by simply concatenating those features. MeTRNN outperforms all other baselines while incorporating both self generated feature and indication information into consideration.

Next, we show the vocabulary for different indication types obtained from the weight matrix W_{mw} learned by MeTLSTM. We randomly select 5 indications from MADE in Table 8. We observe that the vocabulary is closely related to the corresponding indication type. For example, the learned vocabulary for Hodgkin's Lymphoma includes "Hodgkins", "ABVD" (ABVD is a chemotherapy regimen used in the first-line treatment of Hodgkin lymphoma), etc. Later we show that the topics learned by MeTRNN are indeed influenced by the indication feature.

Table 7 shows the vocabulary of randomly selected topics generated by ProdLDA, TopicLSTM and MeTLSTM. Topics learned by ProdLDA and TopicLSTM are similar as they exhibit similar diversity in types of words across topics. Within each topic, we observe more common word, e.g., "deal", "upward" and "med", from ProfLDA and TopicLSTM

Table 7: Top 10 words of 5 topics (randomly selected out of 20) learned by 3 methods. The original words are all in lowercase. Letters are manually capitalized for better interpretation. † or ⋆ indicates that the baseline is implemented by others or ourselves.

Methods	Topic	Vocabulary
ProdLDA†	1	AbdPelvis, Island, Oxymizer, Aids, Acidophilus, Hotline, Things, Greens, CCU, Hypoxemia ...
	2	Laparotomy, Excercize, Striae, Reduce, Cecectomy, Noninflamed, Dipstick Counseled, Transaminitis, DOs ...
	3	Nephrectomy, Amplitudes, Hysterectomy, Stinging, Amplitude, Unimproved, Crease, Prepped, Flexed, Pasty ...
	4	Nonsteroidals, Onethird, Ascertain, Upward, NP, Advancing, Excess, Leaflet, Twothirds, Outflow ...
	5	Deal, Clustered, Proves, Demonstration, Desire, Thinned, Extent, Familysocial, Lobulated, Exclude ...
TopicLSTM⋆	1	Autoimmune, Splenectomy, Marginal, Folic, Reticulocyte, Elbow, Furosemide Calcitonin, Celexa, Losartan ...
	2	Comments, Modified, PO, Medicalsurgical, Laboratorystudies, Communication, SOB, Agree, Temp, Reclast ...
	3	Pediatric, Amitriptyline, Burkitt, Wound, Med, PO, Broviac, Community Headache, Mom ...
	4	Plasmacytoid, Impacted, Badly, Ideal, Priority, Reviews, Fremitus, Expiratory, Accessory, Tactile ...
	5	Testosterone, Lymphoplasmacytoid, Androderm, Bendamustine, Hypogonadism, Acknowledgement, Diltiazem, Kyphoplasties, Alprazolam, Salmonella ...
MeTLSTM⋆	1	eGFR, Antiresorptive, Well, Leery, Equation, MDRDs, SQ, Velcade, Performing, Injuries ...
	2	NP, Amitriptyline, Reports, Pediatric, Burkitt, Palpated, CKD, Kidney, Supervising, Comments ...
	3	Sinuses, Infectious, Transplant, ABVD, Autologous, Acyclovir, Natural, Nasal, Hodgkins, Patient ...
	4	Underwent, Laminectomy, Brachial, Radiation, Intrathecal, Vertebral, Compression, Shoulder, Spondylolisthesis, Insurance ...
	5	Quite, Actually, Breaths, Panic, Attacks, Anxiety, Well, Velcade, Increase, Twice ...

Table 8: Top 10 words of 5 (out of 102) indication types learned by MeTRNN (obtained from weight matrix W_{mw}). The original words are all in lowercase. Letters are manually capitalized for better interpretation.

Indications	Vocabulary
Hodgkin's Lymphoma	Hodgkins, ABVD, Chest, Omeprazole, Chemotherapy, MD, FI, MR, Told, Port ...
Peripheral Neuropathy	Transplant, Peripheral, P, Levels, Neurontin, Marrow, Therapy, Done, Copay, MR ...
Mantle Cell Lymphoma	Cycles, Velcade, Mantel, Location, Therapy, Allogeneic, MD, Positive, Status, Cycle ...
Cellulitis	Cellulitis, Currently, Doxycycline, Redness, Foot, Lymph, Ankle, Anxiety, Rule, Doxazosin ...
Hypercalcemia	Continues, Hypercalcemic, Pamidronate, Radiation, Due, Hospitalization, Weekly, Taking, Schedule, Potassium ...

than from MeTLSTM which is not ideal for capturing unique topics. The topics learned by MeTLSTM emphasize more on different diseases and symptoms as they are influenced by the indication feature. More importantly, such influence mechanized by our proposed MeTRNN improves the modeling performance confirmed by the previous word prediction results.

4.7 Related Work

Context Dependent Neural Language Models. [75] augments contextual information into a conventional RNNLM [74] by adding an extra layer connected to the recurrent unit. The contextual information in this work is obtained by using LDA from a block of proceeding text. TopicRNN [29] extends this idea by integrating a topic model like unit to model the contextual information and the word sequence simultaneously. The topic information is inferred from the document in the bag-of-words representation and is then fed to the recurrent unit to regulate the language modeling in every time step. It uses a variational autoencoder for model inference. [60] introduces an attention-based convolutional neural network to extract semantic topics. [112] incorporates global context of the document obtained from a topic model like unit through a Mixture-of-Experts model design. However, these model do not account for document meta-information for either topic inference or language modeling.

Supervised Topic Models. Author-Topic model [96] assumes words are generated by an author uniformly selected from an observed author list and then a topic selected from a distribution over topics that is specific to that author. [77] models expertise by multiple topical mixtures associated with each individual author. Supervised LDA (sLDA) [71] models document with single label by learning a generalized linear model with an appropriate link function and exponential family dispersion function. Labelled LDA (LLDA) [93] assumes a multi-label document such that each label has a corresponding topic and a document is generated by a mixture of the topics. As an extension to LLDA, Partially Labelled LDA (PLLDA) [94] assigns multiple topics to a label. The Dirichlet Multinomial Regression (DMR) [78] incorporates document meta-information on the prior of the topic distributions with the logistic-normal transformation. [49] introduces a Poisson factorization model with hierarchical document labels. However, these models are *bag-of-words* models that do not consider word ordering.

5 Conclusion

The application scenario and motivation of my dissertation studies are mostly based on the post-market drug surveillance problem. My dissertation studies the problem of exploring, analyzing and modeling various types of sequential data.

Temporal Assertion Analytics. We present the first framework for interactive temporal association analytics. Our **TARA** framework employs a novel evolving parameter space model for pre-generating rules such that near real-time performance is guaranteed for online mining. In a variety of tested cases, **TARA** outperforms the three state-of-the-art competitor techniques, each by several orders of magnitude, while offering a holistic exploration experience supporting new classes of time-variant rule analytics.

In this work we have designed the **MARAS** technology that signals interesting MDAR using contextual information. We defined the non-spurious association that is appropriate for MDAR signals, and proposed the *contrast* measure to find the most severe MDAR signals. When compared with state-of-the-art methods, **MARAS** clearly detects an accurate and diverse set of non-spurious MDAR signals, as confirmed by our case study on FAERS ADR reports data.

Temporal Local Outlier Detection. We present **KELOS** – the first solution for continuously monitoring top-N KDE-based local outliers over sliding window streams. First, we propose the KLOME semantics to continuously capture the *n* points that have the highest outlierness scores in the streaming data. Second, a continuous detection strategy is designed that efficiently supports the KLOME semantics by leveraging the key properties of KDE. Using real world datasets we demonstrate that **KELOS** is 2-6 orders of magnitude faster than the baselines, while being highly effective in detecting outliers from data streams.

Text Modeling and Generation. We propose **MeTRNN** which is a supervised topic compositional neural language model for modeling clinical narratives supported by meta-information. The main idea is to leverage meta-information which hints the semantics of the entire document to regulate the RNN-based language model. We integrate a supervised topic model-like component to allow meta-information to make implicit impact on language modeling via hidden topics. We also propose a black box deep Bayesian inference network for **MeTRNN** which is easily extendable to new models. Through our extensive experiments with several datasets, we show the effectiveness of **MeTRNN** on language modeling as well as the ability of generating useful and meaningful topics.

6 Future Work

I plan to continue my research in the field of data mining and management with a focus on processing and making sense of large scale sequential data presented in real world scenarios. Besides extending and building upon my previous studies, I am also interested in expanding my research into the following directions:

Complex Sequence Modeling. As opposed to word sequence, data sequence in other applications can be complex where each instance is associated with a set of attributes. For example, in a sequence of electronic health records (EHRs) that describes a patient's medical conditions over time, each record is also accompanied by other information such as the patient's demographics, lab test results, admission time, etc. These attributes not only characterize each instance but also encode important behaviour dynamics of the entire sequence. The bipartite structure of attributed sequences poses unique challenges in the modeling tasks. There exist three types of dependencies in an attributed sequence: instance dependencies, attribute dependencies and attribute-sequence dependencies. Thus, learning and capturing these attribute-sequence dependencies are critical for attributed sequence modeling. In addition, the attribute can be of different types. For example, it can be a sequence by itself such as a lab test which is a time series data of a medical measure over time or it can be an image such as a MRI scan of the patient during that particular visit. Incorporating different types of attribute to learn a unified representation of the instance for sequence modeling is also challenging. I plan to study these problems with real world data and evaluate the proposed models in practical downstream applications such as sequence labeling, classification and clustering.

Interpretable Sequence Modeling. In additional to answering the question – *what*, knowing *why* can be more valuable and critical. For example, although machine diagnosis given by the RNNs models learned from massive EHRs may achieve high accuracy in testing phase on historical data, it lacks trustworthiness since the reasoning is a black-box process and is not human interpretable which brings many safety and ethical concerns. Moreover, we as humans cannot benefit or learn from these models to enrich our own knowledge on the subjects. Traditional machine learning algorithms such as *association rule learning* and *decision tree* can give explanation of the decision process via rules applied on the original feature space. Deep sequence models with superior modeling performance by its nature do no provide such reasoning insights. Making sense of these models' internal structures and learned parameters can be challenging. I plan to investigate *attention mechanism* for various RNNs to enable self-explain functionality that presents in understandable terms to a human of how the model operates.

References

[1] Drugs.com. `http://www.drugs.com`. [Accessed 2016-04-20].

[2] National institute of neurological disorders and stroke. `http://www.ninds.nih.gov/disorders/neurotoxicity/neurotoxicity.htm`. [Accessed 2016-10-23].

[3] Openfda. `https://open.fda.gov/drug/event`. [Accessed: 2016-04-20].

[4] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, VLDB 2003, pages 81–92. VLDB Endowment, 2003.

[5] R. Agrawal, M. Mehta, J. C. Shafer, R. Srikant, A. Arning, and T. Bollinger. The quest data mining system. In *KDD*, volume 96, pages 244–249, 1996.

[6] S. M. Al Pascual, Kyle Marchini. Identity fraud: Securing the connected life. 2017.

[7] J. M. Ale and G. H. Rossi. An approach to discovering temporal association rules. In *Proceedings of the 2000 ACM symposium on Applied computing-Volume 1*, pages 294–300. ACM, 2000.

[8] P. Bailis, E. Gan, S. Madden, D. Narayanan, K. Rong, and S. Suri. Macrobase: Prioritizing attention in fast data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD 2017, pages 541–556, New York, NY, USA, 2017. ACM.

[9] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic*, pages 972–986. Springer, 2000.

[10] R. J. Bayardo Jr and R. Agrawal. Mining the most interesting rules. In *SIGKDD*, pages 145–154. ACM, 1999.

[11] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[12] G. S. Birkhead, M. Klompas, and N. R. Shah. Uses of electronic health records for public health surveillance to advance public health. *Annual Review of Public Health*, 36:345–359, 2015.

[13] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[14] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134. ACM, 2003.

[15] D. M. Blei, M. I. Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning*, pages 10–21, 2016.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD 2000, pages 93–104, New York, NY, USA, 2000. ACM.

[19] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets. Using association rules for product assortment decisions: A case study. In *KDD*, pages 254–260, 1999.

[20] R. Cai, M. Liu, Y. Hu, B. L. Melton, M. E. Matheny, H. Xu, L. Duan, and L. R. Waitman. Identification of adverse drug-drug interactions through causal association rule discovery from spontaneous adverse event reports. *Artificial Intelligence in Medicine*, 2017.

[21] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 4(30):891–927, 2016.

[22] L. Cao, M. Wei, D. Yang, and E. A. Rundensteiner. Online outlier exploration over large datasets. In *Proceedings of the 21th ACM SIGKDD*, pages 89–98. ACM, 2015.

[23] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, and E. A. Rundensteiner. Scalable distance-based outlier detection over high-volume data streams. In *Proceedings of the 2014 IEEE International Conference on Data Engineering*, ICDE 2014, pages 76–87. IEEE, 2014.

[24] S. Chaudhuri, H. Lee, and V. R. Narasayya. Variance aware optimization of parameterized queries. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 531–542. ACM, 2010.

[25] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems Workshop on Deep Learning*, 2014.

[27] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5):760–772, 2009.

[28] A. B. Dieng, C. Wang, J. Gao, and J. Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.

[29] A. B. Dieng, C. Wang, J. Gao, and J. W. Paisley. TopicRNN: A recurrent neural network with long-range semantic dependency. *International Conference on Learning Representations*, abs/1611.01702, 2017.

[30] L. Dietz, S. Bickel, and T. Scheffer. Unsupervised prediction of citation influences. In *Proceedings of the 24th international conference on Machine learning*, pages 233–240. ACM, 2007.

[31] G. Dong and J. Li. Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. In *PAKDD*, pages 72–86, 1998.

[32] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[33] C. C. A. S. Edition. *Outlier Analysis*. Springer, 2017.

[34] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5220–5227, 2004.

[35] V. L. et al. Drugbank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, 42(Database-Issue):1091–1097, 2014.

[36] D. M. Fram, J. S. Almenoff, and W. DuMouchel. Empirical bayesian data mining for discovering patterns in post-marketing drug safety. In *SIGKDD*, pages 359–368. ACM, 2003.

[37] E. Gan and P. Bailis. Scalable kernel density classification via threshold-based pruning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD 2017, pages 945–959, New York, NY, USA, 2017. ACM.

[38] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu. Mining frequent patterns in data streams at multiple time granularities. *Next generation data mining*, 212:191–212, 2003.

[39] F. J. G. Gisbert. Weighted samples, kernel density estimators and convergence. *Empirical Economics*, 28(2):335–351, 2003.

[40] M. R. Hacene, Y. Toussaint, and P. Valtchev. Mining safety signals in spontaneous reports database using concept analysis. In *Artificial Intelligence in Medicine*, pages 285–294, 2009.

[41] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, pages 1–12. ACM, 2000.

[42] S. K. Harms and J. S. Deogun. Sequential association rule mining with time lags. *Journal of Intelligent Information Systems*, 22(1):7–22, 2004.

[43] R. Harpaz, H. S. Chase, and C. Friedman. Mining multi-item drug adverse effect associations in spontaneous reporting systems. *BMC Bioinformatics*, 11(S-9):S7, 2010.

[44] C. Heinz and B. Seeger. Cluster kernels: Resource-aware kernel density estimators over streaming data. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):880–893, July 2008.

[45] J. Henry, Y. Pylypchuk, T. Searcy, and V. Patel. Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015. *ONC Data Brief*, 35:1–9, 2016.

[46] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[47] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[48] R. Honda and O. Konishi. Temporal rule discovery for time-series satellite images and integration with rdb. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 204–215. Springer, 2001.

[49] C. Hu, P. Rai, and L. Carin. Non-negative matrix factorization for discrete data with hierarchical side-information. In *International Conference on Artificial Intelligence and Statistics*, pages 1124–1132, 2016.

[50] H. Ibrahim, A. Saad, A. Abdo, and A. S. Eldin. Mining association patterns of drug-interactions using post marketing fdas spontaneous reporting data. *Journal of biomedical informatics*, 60:294–308, 2016.

[51] H. Jin, J. Chen, H. He, G. J. Williams, C. Kelman, and C. M. O'Keefe. Mining unexpected temporal associations: Applications in detecting adverse drug reactions. *IEEE Trans. Information Technology in Biomedicine*, 12(4):488–500, 2008.

[52] R. J. B. Jr., R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. In *ICDE*, pages 188–197. IEEE, 1999.

[53] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing*, 35(3):400–401, 1987.

[54] C. Kiddon, L. Zettlemoyer, and Y. Choi. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, 2016.

[55] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[56] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, volume abs/1312.6114, 2013.

[57] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184. IEEE, 1995.

[58] K. Lang. Newsweeder: Learning to filter netnews. In *International Conference on Machine Learning*, pages 331–339, 1995.

[59] L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer, 2007.

[60] J. H. Lau, T. Baldwin, and T. Cohn. Topically driven neural language model. In *Annual Meeting of the Association for Computational Linguistics*, pages 355–365, 2017.

[61] W.-J. Lee and S.-J. Lee. Discovery of fuzzy temporal association rules. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(6):2330–2342, 2004.

[62] W. Leigh, N. Modani, R. Purvis, and T. Roberts. Stock market trading rule discovery using technical charting heuristics. *Expert Systems with Applications*, 23(2):155–159, 2002.

[63] D. D. Lewis. Reuters 21578 dataset, 1997.

[64] J. Li, D. Maier, K. Tufte, V. Papadimos, and P. A. Tucker. Semantics and evaluation techniques for window aggregates in data streams. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, SIGMOD 2005, pages 311–322, 2005.

[65] Y. Li, P. Ning, X. S. Wang, and S. Jajodia. Discovering calendar-based temporal association rules. *Data & Knowledge Engineering*, 44(2):193–218, 2003.

[66] X. Lin, A. Mukherji, E. A. Rundensteiner, C. Ruiz, and M. O. Ward. Paras: A parameter space framework for online association mining. *Proceedings of the VLDB Endowment*, 6(3):193–204, 2013.

[67] B. Liu, Y. Ma, and R. Lee. Analyzing the interestingness of association rules from the temporal dimension. In *Proceedings of IEEE ICDM*, pages 377–384. IEEE, 2001.

[68] B. Liu, K. Zhao, J. Benkler, and W. Xiao. Rule interestingness analysis using olap operations. In *Proceedings of the 12th ACM SIGKDD*, pages 297–306. ACM, 2006.

[69] F. Liu, A. Jagannatha, and H. Yu. Towards drug safety surveillance and pharmacovigilance: Current progress in detecting medication and adverse drug events from electronic health records, 2019.

[70] C. Lucchese, S. Orlando, R. Perego, and F. Silvestri. Webdocs: a real-life huge transactional dataset. In *FIMI*, 2004.

[71] J. D. Mcauliffe and D. M. Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

[72] A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 786–791, 2005.

[73] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[74] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

[75] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *Spoken Language Technology Workshop*, pages 234–239, 2012.

[76] D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, UAI'08, pages 411–418, Arlington, Virginia, United States, 2008. AUAI Press.

[77] D. M. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *International Conference on Knowledge Discovery and Data Mining*, pages 500–509, 2007.

[78] D. M. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

[79] D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686. ACM, 2006.

[80] B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 412–421. IEEE, 1998.

[81] B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 412–421. IEEE, 1998.

[82] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. LOCI: fast outlier detection using the local correlation integral. In *Proceedings of the 19th International Conference on Data Engineering, March 5-8, 2003, Bangalore, India*, ICDE 2003, pages 315–326, 2003.

[83] E. E. Papalexakis, T. Dumitras, D. H. P. Chau, B. A. Prakash, and C. Faloutsos. Spatio-temporal mining of software adoption & penetration. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 878–885. ACM, 2013.

[84] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[85] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *ICDT*, pages 398–416. Springer, 1999.

[86] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[87] D. Pokrajac, A. Lazarevic, and L. J. Latecki. Incremental local outlier detection for data streams. In *In the proceeding of 2007 IEEE Symposium on Computational Intelligence and Data Mining*, CIDM 2007, pages 504–515. IEEE, 2007.

[88] X. Qin, R. Ahsan, X. Lin, E. A. Rundensteiner, and M. O. Ward. iparas: Incremental construction of parameter space for online association mining. In *Proceedings of the 3rd BigMine*, pages 149–165, 2014.

[89] X. Qin, R. Ahsan, X. Lin, E. A. Rundensteiner, and M. O. Ward. Interactive temporal association analytics. In *EDBT*, pages 197–208, 2016.

[90] X. Qin, L. Cao, E. A. Rundensteiner, and S. R. Madden. Scalable kernel density estimation-based local outlier detection over large data streams. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2019.

[91] X. Qin, T. Kakar, S. Wunnava, B. MacCarthy, A. Schade, H. Q. Tran, B. Zylich, E. Rundensteiner, L. Harrison, S. Sahoo, et al. Mediar: Multi-drug adverse reactions analytics. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1565–1568. IEEE, 2018.

[92] X. Qin, T. Kakar, S. Wunnava, E. A. Rundensteiner, and L. Cao. Maras: Signaling multi-drug adverse reactions. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1615–1623. ACM, 2017.

[93] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Conference on Empirical Methods in Natural Language Processing*, pages 248–256, 2009.

[94] D. Ramage, C. D. Manning, and S. Dumais. Partially labeled topic models for interpretable text mining. In *International Conference on Knowledge Discovery and Data Mining*, pages 457–465, 2011.

[95] S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In *VLDB*, volume 98, pages 368–379. Citeseer, 1998.

[96] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Conference on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[97] S. Sahar. Interestingness preprocessing. In *Proceedings of IEEE ICDM*, pages 489–496. IEEE, 2001.

[98] S. Sahar. Interestingness measures - on determining what is interesting. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pages 603–612. 2010.

[99] M. Salehi, C. Leckie, J. C. Bezdek, T. Vaithianathan, and X. Zhang. Fast memory efficient local outlier detection in data streams. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3246–3260, 2016.

[100] E. Schubert, A. Zimek, and H.-P. Kriegel. Generalized outlier detection with flexible kernel density estimates. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, SDM 2014, pages 542–550. SIAM, 2014.

[101] M.-J. Shih, D.-R. Liu, and M.-L. Hsu. Discovering competitive intelligence by mining changes in patent trends. *Expert Systems with Applications*, 37(4):2882–2890, 2010.

[102] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[103] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[104] W. Spitzer. Importance of valid measurements of benefit and risk. *Medical toxicology*, 1:74–78, 1986.

[105] A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*, 2017.

[106] S. Stilou, P. D. Bamidis, N. Maglaveras, and C. Pappas. Mining association rules from clinical databases: an intelligent diagnostic process in healthcare. *Studies in health technology and informatics*, (2):1399–1403, 2001.

[107] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, VLDB 2006, pages 187–198. VLDB Endowment, 2006.

[108] N. P. Tatonetti, G. H. Fernald, and R. B. Altman. A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports. *JAMIA*, 19(1):79–85, 2012.

[109] N. P. Tatonetti, P. Y. Patrick, R. Daneshjou, and R. B. Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

[110] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[111] K. Verma and O. P. Vyas. Efficient calendar based temporal association rule. *ACM SIGMOD Record*, 34(3):63–70, 2005.

[112] W. Wang, Z. Gan, W. Wang, D. Shen, J. Huang, W. Ping, S. Satheesh, and L. Carin. Topic compositional neural language model. In *International Conference on Artificial Intelligence and Statistics*, pages 356–365, 2018.

[113] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[114] X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and their attributes. In *Advances in Neural Information Processing Systems*, pages 1449–1456, 2006.

[115] L. Wei and J. Scott. Association rule mining in the us vaccine adverse event reporting system (vaers). *Pharmacoepidemiology and drug safety*, 24(9):922–933, 2015.

[116] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

[117] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[118] C. Xiao, E. Choi, and J. Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 2018.

[119] J.-S. Yeh, C.-Y. Chang, and Y.-T. Wang. Efficient algorithms for incremental utility mining. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 212–217. ACM, 2008.

[120] M. J. Zaki. Generating non-redundant association rules. In *SIGKDD*, pages 34–43. ACM, 2000.

[121] Y. Zheng, J. Jestes, J. M. Phillips, and F. Li. Quality and efficiency for kernel density estimates in large data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD 2013, pages 433–444, New York, NY, USA, 2013. ACM.

[122] A. Zhou, Z. Cai, L. Wei, and W. Qian. M-kernel merging: Towards density estimation over data streams. In *Proceedings of the 2003 IEEE International Conference on Database Systems for Advanced Applications*, DASFAA 2003, pages 285–292. IEEE, 2003.