Future Generation Computer Systems 95 (2019) 26–44

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Proactive management of SLA violations by capturing relevant external events in a Cloud of Things environment

Falak Nawaz[a,*], Omar Hussain[a], Farookh Khadeer Hussain[b,*], Naeem Khalid Janjua[c], Morteza Saberi[a], Elizabeth Chang[a]

[a] *School of Business, University of New South Wales (UNSW), Canberra, Australia*
[b] *Centre for Artificial Intelligence, School of Software, University of Technology Sydney, NSW, Australia*
[c] *School of Science, Edith Cowan University (ECU), Perth, Australia*

HIGHLIGHTS

- It justifies the importance of considering external events of interest while determining SLA violations.
- It proposes a systematic framework to capture and process such events for proactive SLA violation identification in a CoT environment.
- It compares results with approaches that do not consider such external events to demonstrate its superiority.

ARTICLE INFO

Article history:
Received 29 July 2018
Received in revised form 13 October 2018
Accepted 15 December 2018
Available online 30 December 2018

Keywords:
Cloud of things
SLA violation
Proactive management
External events
Internal events

ABSTRACT

The cloud of things (CoT) is an emerging paradigm that has merged and combined cloud computing and the Internet of Things (IoT). Such a paradigm has enabled service providers to provide on-demand computing resources from devices spread across different locations for service users to be dynamically connected to them. While this benefits the CoT service providers and users in many ways, it also brings a key challenge of ensuring that the service is delivered according to the promised quality. Failure to ensure this will result in the service provider experiencing penalties of different types and the service user experiencing disruptions. The literature addresses this problem by proactively managing for SLA violations. However, given the geographically dispersed region of a formed CoT service, in this paper we argue that for proactive SLA violation identification, we need specialized techniques that also consider events that are outside the usual control of service providers and users, but will impact the CoT environment and the quality of service. We propose a framework that identifies such external events of interest and ascertains their impact on achieving the service according to the promised quality. We explain the working of our proposed framework in detail and demonstrate its superiority in proactively determining SLA violations as compared to existing approaches.

1. Introduction

The Cloud of Things (CoT) environment which is an amalgamation of the Cloud and the Internet of Things paradigm offers a framework for service users to form on-demand services with different providers that may be located at any location. However, such freedom of choice also brings challenges for service users and service providers. One such challenge common to both is the successful delivery of the service according to the defined QoS parameters, such as response time range, minimum service availability or a certain throughput. These service constraints are usually stated by the service provider in the service offer and are mutually agreed upon by the provider and user in the Service Level Agreement (SLA). However, due to the characteristics of the CoT environment and the services themselves, many services exhibit dynamic QoS variations that result in frequent changes in their behaviours, leading to SLA violations [1–3]. To avoid this, it is necessary to proactively monitor SLAs to prevent violations. As an indication of its importance, the existing literature offers a large amount of work in this area. However, a common shortcoming of the current approaches is that they only focus on events arising from *internal* sources of the CoT environment which are defined as those events that are under the control of the service provider who forms the SLA with the service user. Example of such events are QoS factors or Service Level Objectives (SLOs) such as the service provider delivering a certain level of uptime, a certain level of

[*] Corresponding authors.
E-mail addresses: falak.nawaz@student.adfa.edu.au (F. Nawaz), o.hussain@adfa.edu.au (O. Hussain), farookh.hussain@uts.edu.au (F.K. Hussain), n.janjua@ecu.edu.au (N.K. Janjua), m.saberi@adfa.edu.au (M. Saberi), e.chang@adfa.edu.au (E. Chang).

https://doi.org/10.1016/j.future.2018.12.034
0167-739X/© 2018 Elsevier B.V. All rights reserved.

resources etc. to the service user. In our previous work [4], we demonstrated that the service provider can make better informed SLA violation prevention decisions by considering the internal SLOs and also the nested and hierarchical dependencies present between these SLOs. In this paper, we go beyond this and argue that the service provider to proactively manage SLA violations should also consider another category of events arising from *external* sources apart from just considering the *internal* sources of events. Events from *external* sources of the CoT environment are defined as those that are outside the control of the service provider, but when they occur, they impact on its ability to deliver the QoS as specified in the SLA. Examples of such external sources of events are inclement weather at a CoT node, changed government regulations, short-term political instability, strikes by truck drivers or workers, industrial accidents etc. [5–7] in a dispersed geographic location of the CoT environment which have a direct impact on the fulfilment of the SLA [8]. Thus, it is important for a CoT service provider to also consider events that are external to its boundary but highly related as they will directly influence the internal events while proactively managing SLAs.

With the growth of social media data (e.g. Twitter, Facebook, news sites etc.), there are many different sources from which information related to such external events can be captured on a real-time basis and processed. Twitter is a microblogging platform which has the important characteristic of real-time information, which is categorized into different topics of varying importance. These topics are usually related to various events, which include social events such as sports, parties and political campaigns as well as disastrous events such as accidents, storms, fires and earthquakes. Recent research has shown that tweets can be used to predict various events such as earthquakes [9], crime [10], city traffic events [11], riots [12], strikes and protests [13]. However, to the best of our knowledge, none of the existing research has used Twitter to discover real-time disruptive events to evaluate its impact on CoT applications and predict SLA violations. In this paper, we propose such an approach to detect in real-time external events of interest related to the SLOs of the formed SLA by monitoring tweets and using this information to proactively ascertain the chances of SLA violation. Our focus in this paper is only on Twitter to identify such potential relevant external events but any social media platform can be used by the proposed approach to achieve this aim. The rest of the paper is organized as follows. In Section 2, we discuss the related work in SLA violation management. In Section 3, we present our proposed framework for proactive SLA violation identification by capturing relevant external events. Sections 4–7 explain each module of our proposed framework. Specifically, Section 4 discusses the SLA modelling and identification of relevant external events modules. Section 5 explains the event processing module that extracts information on external events of interest from Twitter feeds. Section 6 discusses the impact assessment module that ascertains the impact of the external event's occurrence on the SLA and Section 7 discusses the reasoning and decision support module that ascertains if a SLA violation is going to occur or not. Section 8 presents the experimental validation of the proposed framework to show its superiority in proactively identifying SLA violations from the existing approaches that do not capture the relevant external events. Section 9 concludes the paper with a discussion on future work.

2. Related work

In this section, we outline the existing research in the area that relates to our proposed approach.

2.1. SLA violation prediction

Many different SLA management models have been proposed to track SLA monitoring and violation prediction in recent years. Recently, in [4], an event-driven approach is developed to predict SLA violations, which combines logic-based reasoning and probabilistic inferencing. Current work extends this work by identifying the external events in a supply chain that impact SLOs and may lead to violations. Another event-driven approach is presented in [14]. However, different from the above work, this approach uses a statistical method for QoS prediction in a service-oriented environment. A similar approach is presented by Leitner et al. [15]. In this work, the authors propose a regression model to predict SLA violations in service compositions. The proposed approach uses a data-driven statistical approach for both instance-level prediction for an ongoing business process instance and forecasting for compliance prediction of future instances. A Bayesian model-based SLA violation prediction is proposed in [16]. The proposed approach uses QoS datasets of cloud services as input to the prediction model, which is based on a naïve Bayesian classifier. In contrast, other approaches for SLA violation prediction use significantly different techniques. Some of the classical and modern heuristics-based time-series forecasting models are listed in [17], which include ARIMA (autoregressive integrated moving average), GARCH (generalized autoregressive conditionally heteroskedastic), regression methods, neural networks, genetic programming and evolutionary computing. Most of the existing violation prediction approaches use historic data to predict future QoS values and none of the approaches addresses the effect of the events that cause these changes in QoS attributes. Fanjiang et al. [2] propose a genetic programming-based QoS forecasting technique based on past QoS variations. The prediction of changing user requirements is handled in [18] by using a Markov chain and best–worst method. Hu et al. [19] propose a personalized QoS prediction approach based on collaborative filtering which combines collaborative filtering with Kalman filtering to improve the shortcomings of ARIMA. An early warning-based framework to detect future violations in SLA is proposed by Hussain et al. [20]. The proposed framework uses exponential smoothing and ARIMA for QoS prediction using the SLO parameters and performance metrics. The runtime QoS values are then compared with the predicted values. Based on the deviation, future QoS values can be predicted over a period of time.

In addition, several studies focus on other aspects of SLAs such as service provisioning, negotiation, privacy and security in the context of cloud federation [21–23]. These aspects are equally important in the context of the CoT environment as well. Messina et al. [24] propose a multi-agent semantic SLA negotiation protocol. This protocol resolves an important concern of negotiation which is understanding the technical terms used by the involved parties. It transforms such technical terms into a common form by sharing the agents' knowledge about these terms among the cloud federation. A model for a per-service-based description of service terms with a particular focus on security-related terms is proposed in [25]. The main feature of the proposed model is to measure, monitor and negotiate security-related service terms quantitatively. Moreover, a framework for the management of a per-service SLA is also presented which is automated and needs no user-intervention. Approaches have also been proposed which employ cloud computing technologies in specific problem areas. For instance, cloud computing technologies have the potential to improve safety, security and privacy, and resource management in intelligent transport systems (ITS) and vehicular ad hoc networks (VANETs). Providing computational services to road users through the VANET-Cloud is discussed in Bitam et al. [26]. This is particularly relevant to the current work in the context of the CoT environment managing a logistics SLA, which is dependent on the safety, security, and reliability of the transport system. A security mechanism to increase reliability of the vehicular safety application is proposed in [27]. This mechanism introduces additional measures to the security procedures both at the transmitter and receiver, which show improvements in terms of safety and QoS metrics.

2.2. Prediction via social media

Recently, many researchers have attempted to use social media to detect or predict earthquakes [9], crime [10], city traffic events [11], riots [12], strikes and protests [13], product sales [28] and the stock market [29]. A recent survey on the predictive power of social media also identified most of the aforementioned application areas [30]. It also discussed a two-level approach consisting of the fine-grained and coarse-grained analysis of social media contents. The fine-grained analysis generates a signal (e.g., emotions, topics) from a social media post or tweet of an individual user. In coarse-grained analysis, a collection of such signals helps domain experts produce insights for predicting the outcome of a real-world event such as elections [31]. In the case of Twitter, a tweet representing some sentiment, emotion, volume, or topic can be considered as signals, while the importance of each of these parameters varies greatly depending on the event and its domain, which makes prediction from social media a real challenge. For example, evaluating the sentiments and emotions of the public towards an electoral candidate is very important in predicting the outcome of an election [31], however the same information may not be as critical in the context of disaster management because the sentiments may be largely negative about the disaster [30]. High-level predictions about real-world events requires not only taking into account a considerably large number of signals but also evaluating present and historical contexts that vary with location and time. Judgments about some real-world event, such as elections, may not only depend on individual signals but may require additional sub-topics such as unemployment, foreign policy etc. to determine the diverse variety of signals. Consequently, state-of-the-art applications, such as Twitris [32] and OSoMe [33], have been developed to process and analyse huge real-time social media data to predict various real-world events. A well-known example of a correct prediction is the 2016 US presidential elections using a social media analytics platform [32]. The researchers conducted real-time predictive analysis making an accurate prediction by analysing state-level signals, such as those from Florida and Ohio, which were considered swing states.

The research work carried out in this paper falls in the area of the extraction of external events from social media and its impact assessment on SLAs, the supply chain and logistics. Little work has been done to investigate the prediction power of social media in this area. Important work in this area was conducted by O'Leary [34] who investigated the capability of social media for its current and potential impact on the supply chain. This work also investigated the use of tags from social media to extend ontology for the supply chain. However, this work neither extracted events from tweets nor performed an impact assessment on SLA which current work examines. Chae [35] proposed a framework for analysing supply chain tweets using the hashtag #supplychain and Twitter analytics. The proposed framework in this work combines descriptive analytics, content analytics (integrating text mining and sentiment analysis) and network analytics (relying on network visualization). This work also highlights the current use of Twitter in supply chain contexts and also develops insights into the potential role of Twitter for supply chain practice and research. However, most of this work relies on the manual interpretation and analysis of tweets while our proposed approach automatically learns the potential external events from incoming tweets using a training set. Moreover, our work uses the predictive power of Twitter for impact assessment on SLA violation and the supply chain which the aforementioned work does not tackle. Furthermore, our proposed framework employs semantic similarity for tweet content analysis as well as considers the spatial and temporal resolution of tweets to identify potential events, which highlights the key differences between the aforementioned studies and the work presented in this paper.

3. Proposed framework for proactive SLA violation identification by capturing relevant external events

In this section, the proposed framework for proactive SLA violation identification by capturing relevant external events is presented. Fig. 1 shows the four modules of the proposed framework.

The roles and responsibilities of each module are as follows:

SLA modelling and identification of relevant external events module: This module takes the SLA document being proactively monitored for SLA violation identification and models the nested and composite relationships present between the SLOs. It then identifies the different relevant external events which, if they occur, will negatively impact the SLOs and the SLA in general. The working of this module is explained in Section 4.

External Event Processing Module: After the external events relevant to the SLA are identified, this module takes a Twitter stream as input and annotates entities in each tweet using a training model. Each annotated entity of a tweet is assigned a relevant external event from the risk events corpus based on its semantic similarity with them. Tweets whose annotated events match with those identified as external events of interest to the SLA are further filtered based on their location and timestamp. If these factors are satisfied, then the identified relevant external event is assigned a probability of occurrence which shows a measure of its strength based on the frequency of that event in the Twitter stream. The working of this module is explained in Section 5.

Impact Assessment Module: The impact assessment module ascertains the impact of relevant external events on the SLOs. In other words, this module determines the variation in a SLO from its QoS values defined in the SLA, due to the occurrence of the relevant external event according to its severity. The impact is determined by considering the different types of impacts, such as financial, service, reputation etc. The working of this module is explained in Section 6.

Reasoning and Decision Support Module: The occurrence of any event initiates the framework's ability to capture the QoS values related to the SLOs being monitored and store them in QoS repository. The event may be triggered due to some external factors, such as the occurrence of a strike at a particular location or it can occur due to internal factors, such as QoS performance variation or degradation at different points during the service execution. In case of the event being an external one, its probability of occurrence and its impact on the SLOs are determined. Once the probability and the impact of the relevant external event on the SLOs have been determined, this module ascertains the final state of the SLA using the composite relationships modelled in the first module. Logical and probabilistic reasoning is used to predict if any potential SLA violation is going to occur or not. The working of this module is explained in Section 7.

The sequence of the steps for proactive SLA violation identification by capturing relevant external events is shown in Fig. 2. In module 1, based on the SLA, the service user should first identify which guarantee terms (GTs) of the SLA will be impacted by external events. These identified relevant external events are saved in the *risk events repository*. Next, in module 2, real-time information related to the CoT setup is analysed and associated events are extracted. These extracted events are analysed to see if they match

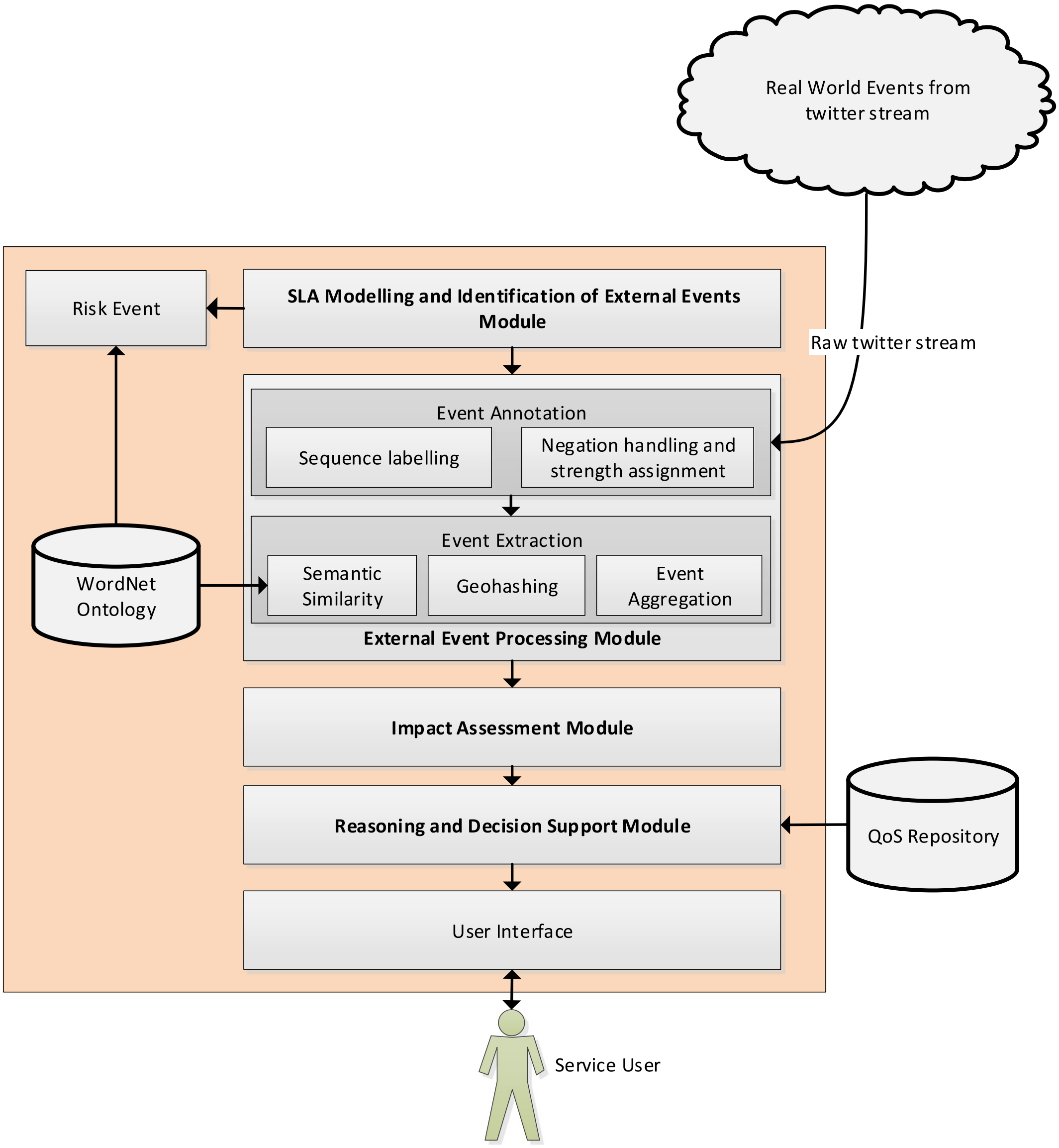

Real World Events from twitter stream
Raw twitter stream
Risk Event
SLA Modelling and Identification of External Events Module
Event Annotation
Sequence labelling
Negation handling and strength assignment
Event Extraction
Semantic Similarity
Geohashing
Event Aggregation
External Event Processing Module
WordNet Ontology
Impact Assessment Module
Reasoning and Decision Support Module
QoS Repository
User Interface
Service User

Fig. 1. Different modules for proactive SLA violation identification by capturing relevant external events.

with those identified as external events of interest to the SLA by analysing their characteristics, such as semantically matching event terms, location, time etc. If a match with the relevant external event is found, then in module 3, its probability of occurrence along with its severity is computed and their impact on the SLOs is quantified. In module 4, the impact of the occurrence of the relevant external event on the final state of the SLA is determined by using logical and probabilistic reasoning.

4. SLA modelling and identification of relevant external events module

As described previously, the prerequisite for the external event processing module to start working is to identify the relevant external events that can impact the SLOs of the SLA. This is achieved in the *SLA modelling and identification of relevant external events module* which has four steps. The first step is to model the SLA and identify the nested relationships between the SLOs of the SLA. The second step is to formalize the states in which the SLOs can be in at different points of time to determine the final state of the SLA. The third step is to determine the composite relationships present between the different SLOs that will help determine the state of the SLA. The fourth step is to identify the different relevant external events that can impact the SLOs of the SLA. Each step is explained in detail below.

4.1. Modelling the SLA according to the nested relationships present between the SLOs

In this section, we explain the preliminaries of modelling a WS-Agreement [36] SLA in multi-valued logic form. A WS-Agreement SLA consists of one or more SLOs. Each SLO is referred to as a guarantee term (GT) that needs to be satisfied as defined in the SLA to prevent the violation of the overall SLA. The GTs can be in a hierarchical or nested structure in the SLA. For example, Fig. 3 shows the relationships between the GTs that are extracted from a textual WS-Agreement SLA. GT_2 and GT_3 are nested under GT_1 and GT_5 and GT_6 are nested under GT_4 but GT_7 is not nested or dependent on other GTs.

Each GT is further classified either as an individual guarantee term (IGT) or composite guarantee term (CGT) depending upon its relationships with the other GTs in the SLA. These are explained further as follows:

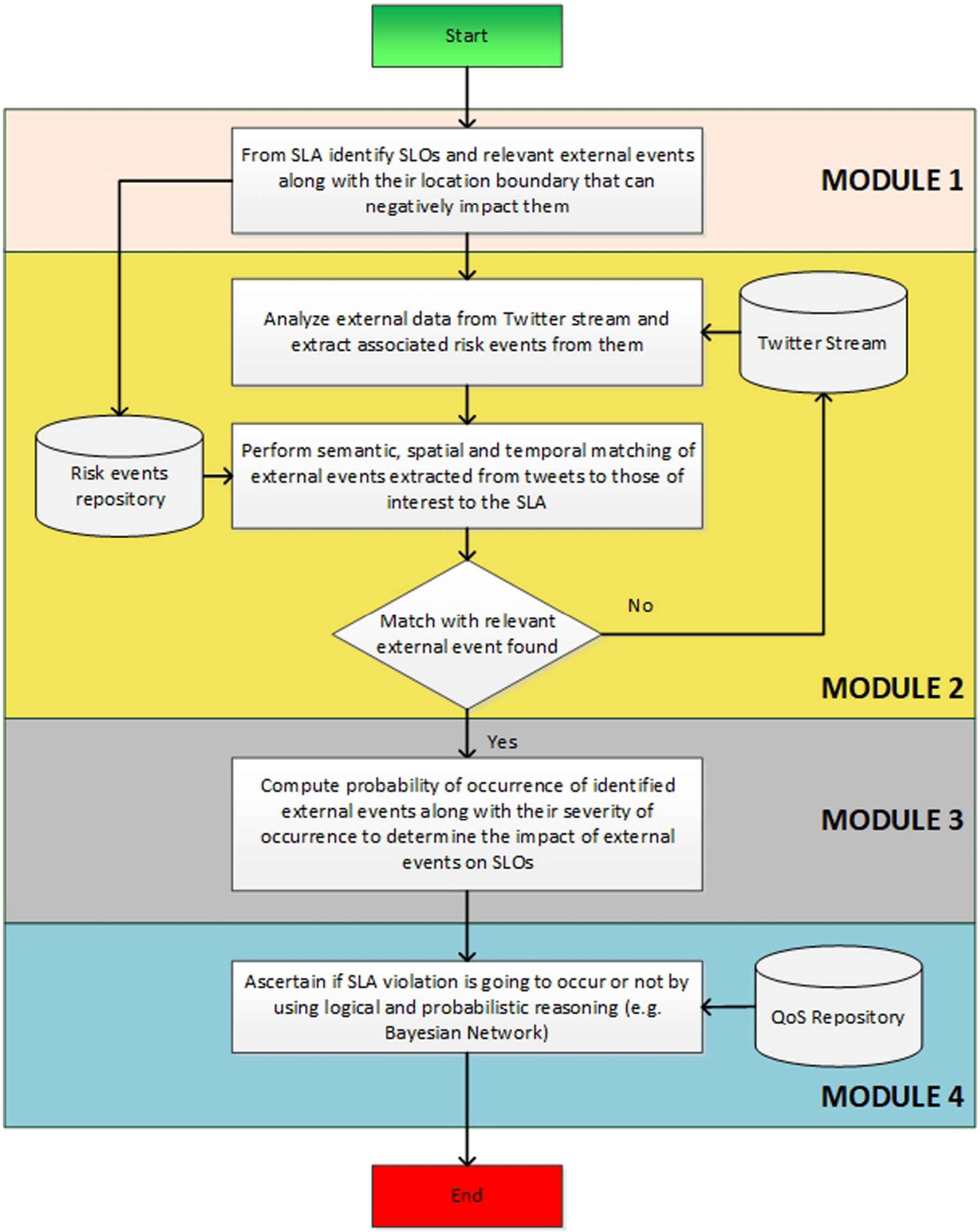

Start
From SLA identify SLOs and relevant external events along with their location boundary that can negatively impact them
MODULE 1
Analyze external data from Twitter stream and extract associated risk events from them
Twitter Stream
Risk events repository
Perform semantic, spatial and temporal matching of external events extracted from tweets to those of interest to the SLA
Match with relevant external event found
No
MODULE 2
Yes
Compute probability of occurrence of identified external events along with their severity of occurrence to determine the impact of external events on SLOs
MODULE 3
Ascertain if SLA violation is going to occur or not by using logical and probabilistic reasoning (e.g. Bayesian Network)
QoS Repository
MODULE 4
End

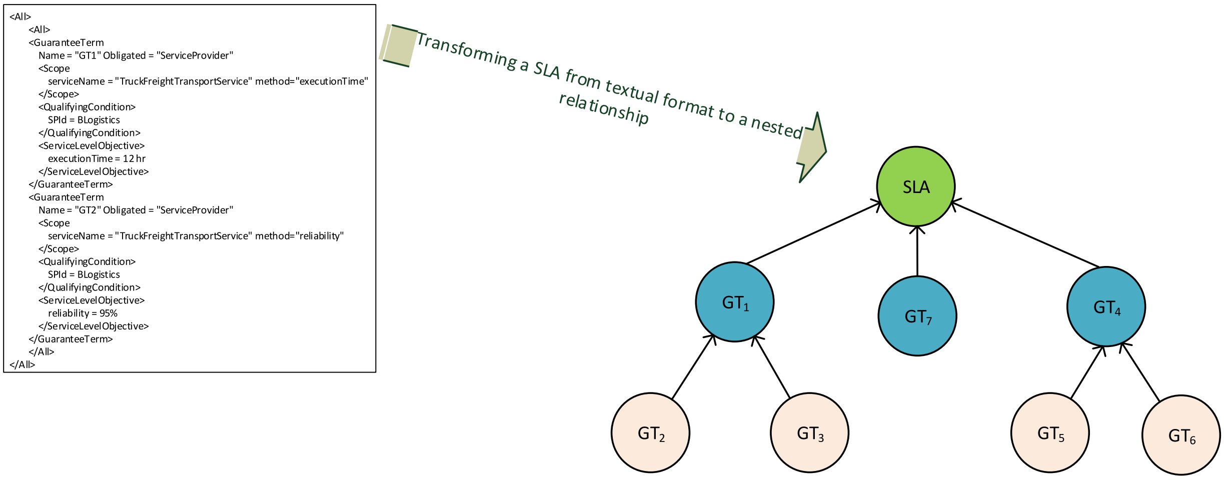
Fig. 2. Flow chart showing the sequence of steps in the proposed system.

1. *Individual Guarantee Term (IGT).* An IGT is an individual guarantee term in the SLA. In other words, these are GTs that are not dependent on other GTs for their final state. For example, from Fig. 3, GT_2, GT_3, GT_5, GT_6 and GT_7 are IGTs.
2. *Composite Guarantee Term (CGT).* A CGT is a guarantee term which is the result of the combination of two or more IGTs. In other words, these are GTs that are dependent on other IGTs. For example, from Fig. 3, GT_1 and GT_4 and the SLA are CGTs.
3. *Guarantee Term (GT).* As previously mentioned, each GT is an SLO of the SLA. The GT can either be a CGT or IGT at different points in time. For example, from Fig. 3, when IGTs, GT_2 and GT_3 are being combined to ascertain the state of GT_1, then GT_1 is a CGT. But when GT_1, GT_7 and GT_4 are being combined to ascertain the state of the SLA, then each of them is an IGT. So, hereon in this paper, to avoid confusion when determining the state of a CGT, an IGT or CGT is represented simply as a GT.

4.2. Defining the possible states in which the GTs can be at different points in time

When the chance of SLA violation is being determined, the commitment of each GT to its defined QoS value needs to be ascertained. This is done by comparing its defined QoS constraints in the SLA with the recently monitored QoS values. This run-time commitment of each GT to its defined constraint values is measured and expressed in one of the following *three* states [4]:

1. *Satisfied (S).* If the runtime QoS of the GT is within the defined constraints of the SLA, then it is represented as *satisfied*.



```
<All>
    <All>
    <GuaranteeTerm
      Name = "GT1" Obligated = "ServiceProvider"
      <Scope
        serviceName = "TruckFreightTransportService" method="executionTime"
      </Scope>
      <QualifyingCondition>
        SPId = BLogistics
      </QualifyingCondition>
      <ServiceLevelObjective>
        executionTime = 12 hr
      </ServiceLevelObjective>
    </GuaranteeTerm>
    <GuaranteeTerm
      Name = "GT2" Obligated = "ServiceProvider"
      <Scope
        serviceName = "TruckFreightTransportService" method="reliability"
      </Scope>
      <QualifyingCondition>
        SPId = BLogistics
      </QualifyingCondition>
      <ServiceLevelObjective>
        reliability = 95%
      </ServiceLevelObjective>
    </GuaranteeTerm>
    </All>
</All>
```

Transforming a SLA from textual format to a nested relationship
SLA
GT$_1$
GT$_7$
GT$_4$
GT$_2$
GT$_3$
GT$_5$
GT$_6$

Fig. 3. Different levels of nested relationships between GTs of an SLA.

2. *Violated (V).* If the runtime QoS of the GT is not within the constraints as specified in the SLA, then it is represented as *violated*.
3. *Uncertain (U)*: If a GT can neither (a) be evaluated as satisfied nor violated as it cannot be measured [37] *or* (b) is in the range that is near to the defined QoS constraint, then it is represented as *uncertain*. The reason for not being able to measure a GT is because dynamic QoS monitoring from the external sources, such as online third party monitoring, may represent situations such as missing QoS or imprecise QoS [37]. Also, situations where the QoS value of a GT is near the defined constraint are too represented to be in an *uncertain* state [4].

GTs can either be in a *satisfied*, *violated* or *uncertain* state during SLA execution. For GTs like GT$_1$ and GT$_4$ of Fig. 3, their final states are dependent on the composition relationships between the different underlying GTs. These compositor relationships are namely *All Compositor, OneorMore Compositor and ExactlyOne Compositor*, as explained in the next sub-section.

4.3. Different possible compositor relationships between the GTs

All, OneorMore and *ExactlyOne* are compositors to join one or more GTs (for example GT$_2$ and GT$_3$ of Fig. 3) and find the state of the resulting GT (for example GT$_1$). The result from these compositors is dependent on the states of the individual GT$_s$ (for example GT$_2$ and GT$_3$) and the intended state of the resulting GT (for example GT$_1$) to be achieved, whereupon, in this work, we consider it to be satisfied. With these assumptions, the output of each compositor is as follows:

1. *All Compositor:* An *All compositor* combines multiple GTs and represents the CGT as being in its intended state (*satisfied* state) when all of its GTs are in a *satisfied* state and/or when they are in either a *satisfied* or *uncertain* state. The CGT is in a *violated* state if even one of its GTs is in a *violated* state. The CGT is in an *uncertain* state if the GTs are all in an *uncertain* state.
2. *OneOrMore Compositor:* An *OneOrMore compositor* combines multiple GTs and represents the CGT as being in its intended state (*satisfied* state) when at least one of the GTs is in a *satisfied* state. The CGT is in a *violated* state when none of its GTs are in a *satisfied* state and at least one GT is in a *violated* state. The CGT is in an *uncertain* state when all of its GTs are in an *uncertain* state.
3. *ExactlyOne Compositor:* An *ExactlyOne compositor* combines multiple GTs and represents the CGT as being in its intended state (*satisfied* state) when only one of its GTs is in a *satisfied* state. The CGT is in a *violated* state when either more than one GT is in a satisfied state or there is no GT in a *satisfied* state. The CGT is in an *uncertain* state when all of its GTs are in an *uncertain* state.

The state of the resulting GT using the *All, OneOrMore* and *ExactlyOne* compositors on two GTs is shown in Table 1. For example, in rule 1, when both GT$_s$ are in a *satisfied* state, the *All* and *OneOrMore* compositors give the state of the resulting GT as *satisfied.* However, the *ExactlyOne* compositor gives the resulting GT state as *violated* as, according to its definition, for it to be in a *satisfied* state, exactly one GT must be in *that* state. For a more detailed explanation of this logic with examples, readers are referred to our previous work [4].

4.4. Identifying the different relevant external events that can impact the GTs of the SLA along with their location boundary

The objective of this step is to identify the relevant external events that have the potential to impact on the SLA GTs.

Identifying which external events impact on the GTs of the SLA is a two-step process.

- The first step is to manually map the GTs to the external risk events. The logic here is that if the external risk events occur, then they will negatively impact the GTs to which they are mapped. These external risk events (shown in the *External risk events* column of Table 2) are extracted from the WordNet ontology and we assume that an organization has predefined them under the risk category, as shown in Table 2. The aim of the SLA manager in this step is to link these external risk events (such as protests, incidents, natural disasters, labor disputes etc.) to the GTs they will negatively impact . Similarly, external risk events such as accidents, labor strikes, rainstorms etc. can negatively impact a GT relating to the transportation risk category.
- The second step is to assign a location boundary to each external risk event identified as impacting a GT in step 1. For example, if in step 1, an external risk event such as a labor strike is identified as negatively impacting a GT, an important point to note here is that this holds true if such an external risk event happens around a geographic location where the organization in question is located. So, in step 2, we assign the location boundary in kilometres which signifies that if

Table 1
Truth values of compositor elements.

Rule #	GT 1	GT 2	Resulting GT (All)	Resulting GT (OneOrMore)	Resulting GT (ExactlyOne)
1	S	S	S	S	V
2	S	V	V	S	S
3	S	U	S	S	S
4	V	S	V	S	S
5	V	V	V	V	V
6	V	U	V	V	V
7	U	S	S	S	S
8	U	V	V	V	V
9	U	U	U	U	U

Table 2
Potential relevant external event (EE) types.

Risk category	*External risk events*
Supplier risk	protests, terror incidents, political instability, tournaments, economic issues, natural disasters, epidemics, labor strikes, marathons, climate change, pandemics, concerts, fires, quality issues.
Transportation risk	road construction/repairs, accidents, logistic provider carrier capacity, transportation cost changes, airport proximity, labor strikes, weather, rainstorms, deluges.

the external risk event (identified in step 1) occurs within a radius of defined kilometres of a particular location, then it will negatively impact the GT. At the end of this step, as shown in Fig. 4, we will have the GTs of the SLA linked with *relevant external risk events,* each represented by a tuple *{EE, locbdy}* where *EE* defines the *relevant external risk event* that can impact them and *locbdy* defines the location boundary with the radius in kilometres, defined in terms of latitude and longitude.

5. External event processing module

In this section, we explain the proposed methodology for the impact assessment of an external event for SLA violation prediction. Social media (e.g. Twitter) is a useful tool to capture information about real-world events happening around the world. An important characteristic of Twitter is the real-time nature of tweets, which can be used to detect the occurrence of real-time events such as earthquakes [9]. But the challenge of extracting such events from a Twitter stream is that it is an open domain textual data. This means it may not follow any rules of grammar, which makes it harder to process using traditional techniques. So, our approach is to first analyse the Twitter stream to identify what events they relate to and then see if they match with the relevant external events that impact the GTs. To do this, we process the external events from a Twitter stream of data in four steps: (i) event annotation of a Twitter data stream (ii) identifying the presence of negation in tweets and determining its strength (iii) extracting tweet events and matching them with the identified relevant external events (iv) filtering the matched events according to their location boundary and (v) event aggregation and the probability of the occurrence of a relevant external event.

5.1. Event annotation of twitter data stream

In this step, different entities are annotated and identified from tweets. For this purpose, we follow a similar approach to that presented in Anantraman et al. [11] with some modifications. A tweet *($tweet_n$)* is considered as a sequence of tokens. The tokens of a tweet are obtained by using the *token($tweet_n$).* Each token of a tweet is assigned a label and a suffix (also known as a tag), using a variant of BIO (Beginning, Inside, Outside) notation [38]. For a single word entity, the label *B-suffix* is used. For a multi-word phrase, label *B-suffix* is used for the first word while *I-suffix* is used for all the remaining words in the phrase. The suffix represents the nature of the phrase's word. In other words, a suffix represents if the phrase's word is a location or event or time. For example, if a location entity phrase such as 'Northcott Drive Canberra' is annotated using the BIO notation, we get 'Northcott' as *B-Location,* 'Drive' as *I-Location* and 'Canberra' as *I-Location*. Similarly, an event phrase can also be annotated using B-Event and I-Event tags. For example, as shown in Fig. 5(a), the phrase 'Labor strike is tomorrow' is annotated as 'Labor' as *B-Event,* 'strike' as *I-Event',* 'is' as *O,* and 'tomorrow' as *B-Time.* If a phrase's word does not refer to an entity (such as events or locations), then the O-suffix is used to label it. In general, the tag set is represented as, $\text{Tag}_{\text{set}} = \{$B-Event, I-Event, B-Location, I-Location, B-Time, I-Time, O$\}$.

A bidirectional long short-term memory (Bi-LSTM) combined with conditional random field (CRF) is used to annotate each tweet's token in our approach. A Bi-LSTM-CRF model is a combination of (i) *Bi-LSTM*, which can use both past and future input tokens for event annotation, and (ii) *CRF*, which is an undirected graphical model containing sentence level tag information. The Bi-LSTM-CRF is trained on a set of tokens as events extracted from the Twitter training dataset (given in [11]) and Wordnet ontology concepts which represent potential external risk events (shown in Table 2 and described in Section 4.4). As shown in Fig. 5(a), a CRF model's output is a set $S = \{tokens(tweet_n), Tag_{set}\}$ where *tokens($tweet_n$)* represents a word extracted from a tweet and Tag_{set} represents its annotated BIO-suffix. The CRF model computes scores (represented as f_θ) for all possible combinations of the *input variable*. The *input variable* combination represents the dependency which is present between (a) neighbouring tags *(tag_i,* tag_{i+1}*)* (b) tags and sequence *(tag_1, $token_1$), …,(tag_i, $token_i$), …,(tag_m, $token_m$)* where $\text{tag}_\text{i} \in \text{Tag}_\text{set}$ and $\text{token}_\text{i} \in$ *tokens($tweet_n$)* [11]. The score of the *input variable* combination (e.g. (tag_i, tag_{i+1})) is denoted as $f_\theta(tag_i, tag_{i+1})$ and it represents the number of times tag_i is encountered before tag_{i+1}. This allows the CRF model to predict the most suitable tag for a token in a sentence. Combining the CRF with Bi-LSTM component enhances the annotation process by efficiently using the past and future tags to predict the current tag. Bi-LSTM-CRF computes scores f_θ along with transition score $[A]_{i,j}$ to model transition from *i*th state to *j*th state [39]. The function f_θ becomes $f'_\theta = f_\theta \cup [A]_{i,j}$, which improves the accuracy of annotation. We trained the Bi-LSTM-CRF model by using Parts of Speech (POS)-tagged data of different tweets. For each tweet, this model returns the input twitter data stream in the form of tuple (*e, t, l*) where *e* consists of event terms, *t* represents the time or day, and *l* represents the location. An example of our objective of annotating an event with time of occurrence in a tweet is shown in Fig. 5. Our primary objective in this step is to accurately identify events. If an event is identified but its location and time are not

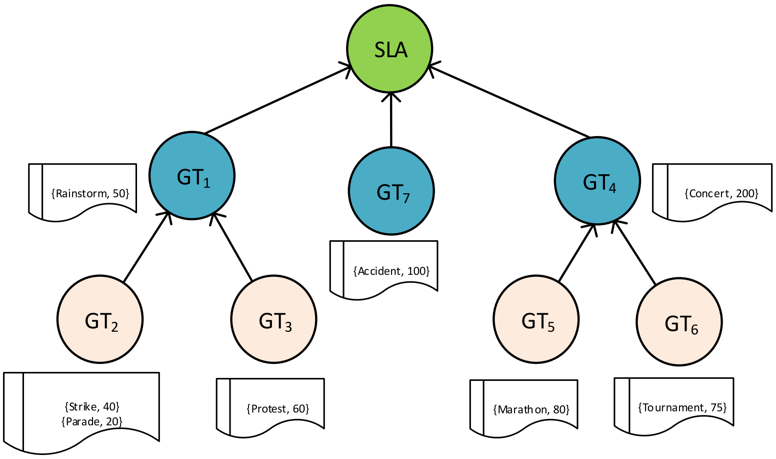
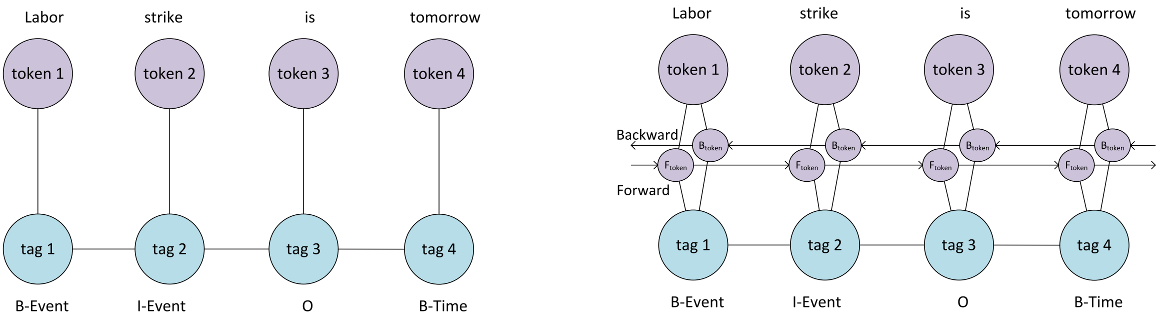


Fig. 4. Different external events and the GTs they impact along with the location boundary.



(a) CRF

(a) Bi-LSTM CRF

Fig. 5. Event annotation models for Twitter data stream.

found from the tweet, then we identify them from the GPS location and the timing of the tweet respectively. This is done in the event extraction step.

5.2. Identifying the presence of negation in tweets and determining the emotion strength of annotated events

Once the tweets are annotated with event terms, they are first checked for the presence of negation. This is important to determine if an event is actually happening. For example, the tweet *’no strike tomorrow’* represents that the event ‘strike’ will not happen tomorrow. In such a case, this event is not considered further by our framework as it will not happen and does not pose a risk that will lead to SLA violation. Negation in linguistics represents the process of inverting the sentiment of a word. Some examples of negation words in the English language are *‘not’*, *‘no’*, *‘n’t’*, *‘none’*, *‘never’*. Moreover, some verbs are implicitly used to represent negation when used with specific nouns e.g. *strike cancelled*. So, in this check, the annotated event terms from step 5.1 are used to check if negation is present in the tweets. If so, then such events are assigned a strength of -1 and they, along with the tweets they are coming from are not considered further. However, before completely disregarding such events, the timestamp t' of their tweets is noted. This is because in the set of events that are processed in a series, there may be a possibility that a similar event to the one which is assigned a strength of -1 (from tweet of timestamp t') is encountered again from a tweet at timestamp t''. In such cases, it is checked that if $t'' > t'$, then it means that the event which was discarded is mentioned again in a later tweet; and in the case of no negation present in the tweet with timestamp t'' then that event is not discarded and is considered further. If $t'' < t'$, then it means that the event which is discarded due to the presence of negation at timestamp t' is the most recent one, and it remains discarded from further analysis.

Events from tweets which do not have any negation are assigned a strength value of $+1$. But assigning this value to events that do not have negation associated with them does not capture the intensity of emotion or sentiment present in them. For example, the following two annotated events of ‘strike’ and ‘maybe strike’ will be assigned the same strength value of $+1$. This is not correct as the second event involves some uncertainty, which needs to be captured to adequately represent its strength value [40]. In our approach, we do this by utilizing adverbs. Adverbs can be categorized into five types according to [41]: (i) *affirmation*: e.g. exactly, certainly (ii) *intensifier* e.g. surely, extremely (iii) *doubt*: e.g. seemingly, apparently (iv) *weakener* e.g. slightly, weakly and (v) *negation*: e.g. hardly, barely. Once an initial sentiment strength of $+1$ is assigned to an event, its value is increased to $+0.5$ if an *affirmation* or *intensifier* adverb is encountered in this tweet. For instance, the event strike in a tweet *‘strike is surely tomorrow’* gets a strength value of $1 + 0.5 = 1.5$. Similarly, if a *doubt* or *weakener* adverb is encountered in the tweet, then the sentiment strength of the annotated event is decreased by -0.5. For instance, *‘Probably strike is tomorrow’* gets a sentiment strength value of $1 - 0.5 = 0.5$.

5.3. Extracting tweet events and matching these with identified relevant external events

In this step, annotated events from processed tweets are extracted and analysed to determine if they match with the external events of interest to the SLA. In other words, the events of each processed tweet from Section 5.1 represented by *e* from the tuple (*e, t, l*) are extracted with the objective to match it to its most *relevant external event* (*EE*). Matching *e* with *EE* also addresses the problem of different tweets using different terms to refer to the same event. For example, some tweets may use the term *typhoon* to refer to a tropical storm *(EE)* whereas some may refer to it as a *hurricane.* Similarly, *rainstorm* and *deluge* can be used to refer to the same event, *floods (EE).* Once *e* is matched in the WordNet ontology, by measuring the semantic similarity of *e* with *EE,* similar event terms *e* can be clustered into the same relevant external event *EE*. The level of match is determined by using an ontology-based approach [42] that considers the taxonomical categorization of concepts represented in the WordNet ontology to determine the semantic distance between any two concepts. We consider that the set of taxonomical features associated with a concept includes all the upper hierarchical concepts that subsume it. More specifically, given two concepts, this approach considers the degree of overlap between their common taxonomical features as proportional to their similarity and the degree of disjunction between their non-common taxonomical features as a function of their dissimilarity. Using the notations introduced by [42], the normalized similarity is computed as a function of dissimilarity between concept *a* and *b* according to their taxonomical features, which is given as follows:

$$S(a,b) = (1 - \log_2\left(1+ \frac{|\varnothing(a)\backslash\varnothing(b)| + |\varnothing(b)\backslash\varnothing(a)|}{|\varnothing(a)\backslash\varnothing(b)| + |\varnothing(b)\backslash\varnothing(a)| + |\varnothing(a)\cap\varnothing(b)|}\right)) \quad (1)$$

where: $|\varnothing(a)\backslash\varnothing(b)|$ represents the set of differential (non-common) features of *a* with respect to *b*,

$|\varnothing(b)\backslash\varnothing(a)|$ represents the set of differential features of *b* with respect to *a*, and

$|\varnothing(a)\cap\varnothing(b)|$ represents the set of common features of *a* and *b*.

For example, Fig. 6 shows a portion of the WordNet ontology where the similarity between the concepts of *strike* and *protest* is computed as:

$$\begin{aligned}\varnothing(\mathit{strike}) = \{&\mathit{strike}, \mathit{job\ action}, \mathit{boycott}, \mathit{protest}, \mathit{resistance},\\ &\mathit{group\ action}, \mathit{human\ action},\\ &\mathit{event}, \mathit{psychological\ feature}, \mathit{abstraction}, \mathit{entity}\}\\ \varnothing(\mathit{protest}) = \{&\mathit{protest}, \mathit{resistance}, \mathit{group\ action}, \mathit{human\ action},\\ &\mathit{event}, \mathit{psychological\ feature}, \mathit{abstraction}, \mathit{entity}\}\end{aligned}$$

From these sets of features, the similarity of these concepts using Eq. (1) is:

$$S(\mathit{strike}, \mathit{protest}) = (1 - \log_2\left(1 + \frac{3+0}{3+0+8}\right)) = 0.65$$

A similarity measure close to 1 represents similar concepts whereas a similarity measure close to 0 represents dissimilarity. Usually, concepts with less semantic distance, such as subsumption (parent–child) or siblings' relationship, in the WordNet ontology are semantically similar concepts. Therefore, we assume concepts with such relationships are semantically similar and group them into one relevant external event. After events terms *e* are identified from tweets, their similarity to all the *EEs* is computed and if their similarity is above a threshold value β of 0.6, then they are considered to be related to the *EE*. In the case of an *e* having a similarity match value greater than the threshold with more than one *EE*, then *e* is matched to the *EE* with the highest similarity match.

The process of matching *e* with *EE* works well if *e* is a single word event. In cases where *e* is multi-word such as *labor strike* then challenges occur as the WordNet ontology is not fully multi-word. To resolve this issue, we breakdown *e* into individual words and try to find a match for each word of *e* with the nodes or concepts of the WordNet ontology. If there is a match, then the multi-word *e* is matched to that concept of the WordNet ontology. For example, if *e* is *labor strike,* then it is broken down into *labor* and *strike.* The word strike matches with a concept of the WordNet ontology as the word and in this case, *e* which is *labor strike* is matched to the strike concept of the WordNet ontology. In a case where each individual word of *e* consists of WordNet concepts, then the similarity of each concept to its *EE* is determined and the multi-word *e* is matched to that concept which has the highest similarity match with *EE*. Another challenge is how to handle two similar concepts, for example, *protest* and *strike* which have both been identified as *EE* by the cloud service manager. This is addressed by assigning a priority to each of them by the cloud service manager. The one with the higher priority can used to cluster similar event terms that come under it. Alternatively, the clustering of such similar event terms can be prevented in the similarity matching process by increasing the threshold value β. Even if two different events are clustered into a single relevant external event in this step, location-based filtering can filter out events occurring at different locations. This is explained in the following section.

5.4. Filtering matched events according to their location boundary

To determine if an event *e,* which is matched to a relevant external event, is occurring within the radius of the defined location boundary *locbdy,* we use Geohash. Geohash splits a specified region into grids, as shown in Fig. 7 of the greater Sydney area. It converts the geolocation information to an alphanumeric hash value, which is unique to each grid. The size of the grid is determined by the geohash length which ranges from 1 to 8 characters. The GPS location of the tweet, which mentions event *e,* is used to find out the grid to which it belongs. A unique identifier of the grid is assigned to the location information *(l)* of the tweet in the tuple (*e, t, l*). This identifier is then used to filter events *e* based on their location and determine the number of tweets referring to an external event *EE* within and outside the location boundary *locbdy* of *EE.* As a rule of thumb, we choose a suitable size of the geohash depending upon the radius of *locbdy* of *EE.*

5.5. Event aggregation and probability of occurrence of relevant external event

The next step is to aggregate all *e* related to *EE* and ascertain *EE*'s probability of occurrence. It is important to note that the probability of occurrence of *EE* is computed if the number of tweets mentioning it is greater than a defined threshold Ω. The logic behind this is that if there is only one tweet mentioning a relevant external event *EE*, then it may not be significant to consider as compared to another *EE* for which there are, say, more than 20 tweets. We consider the value for Ω as 5 in our approach. Eq. (2) aggregates and computes the probability of occurrence of *EE*:

$$\begin{aligned}Pr(EE) = {}& AvgSim(e \in EE)\\ &* \left(\frac{[\mathit{Strength\ of\ tweets\ within\ locbdy}]}{[\mathit{Total\ strength\ of\ tweets\ related\ to\ EE}]}\right),\\ &\mathit{if}\ Ftweets(EE) > \Omega \end{aligned} \quad (2)$$

where $Pr(EE)$ represents the probability of occurrence of relevant external event *EE,*

$AvgSim(e \in EE)$ represents the average of the similarity value of various *e* related to *EE,*

$[\mathit{Strength\ of\ tweets\ within\ locbdy}]$ represents the strength of tweets for *e* related to *EE* which are within the location boundary *locbdy*,

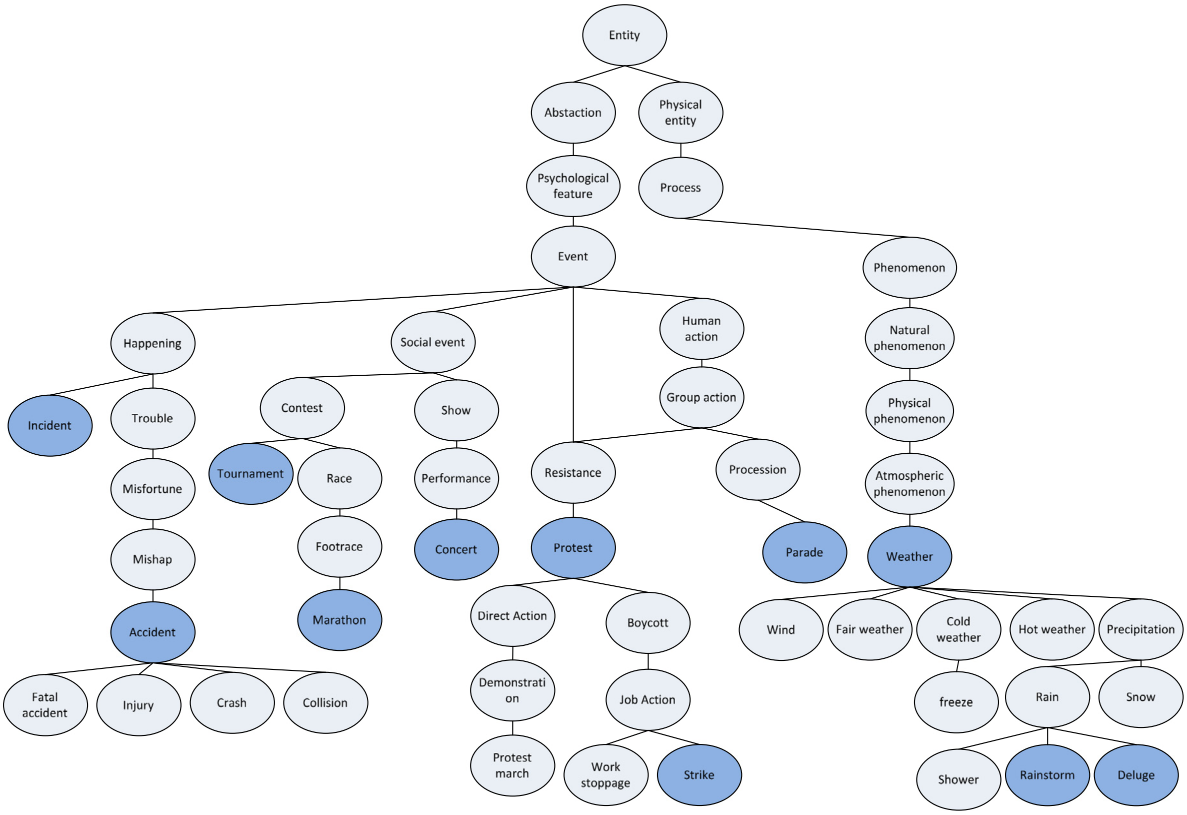


Fig. 6. Portion of WordNet Ontology showing different words and their relationships.

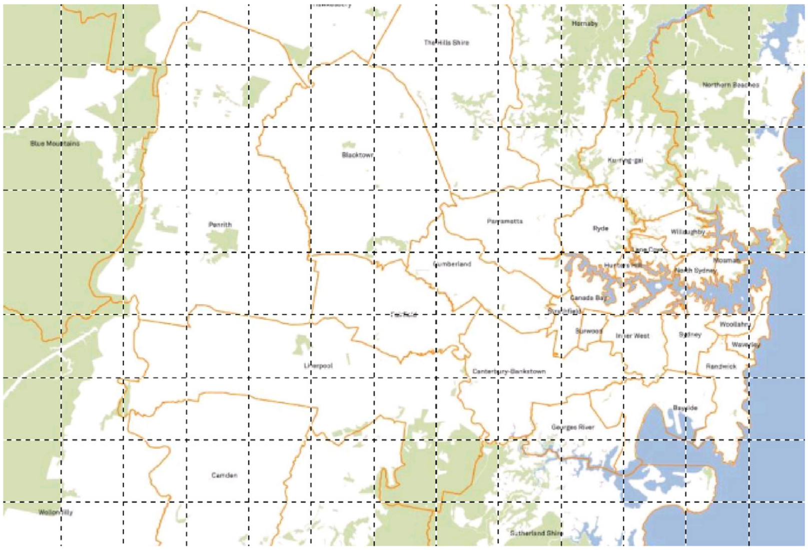


Fig. 7. Geohash of the greater Sydney area.

$[Total\ strength\ of\ tweets\ related\ to\ EE]$ represents the total strength of tweets for all *e* clustered into *EE*,

$Ftweets\,(EE)$ represents the total number of tweets with event *e* clustered into *EE*.

To compute $\left(\frac{[Strength\ of\ tweets\ within\ locbdy]}{[Total\ strength\ of\ tweets\ related\ to\ EE]}\right)$, strength value of every tweet clustered into *EE* is added. If most of the tweets are outside the *locbdy*, it results in less probability of the occurrence to be determined for the relevant external event *EE* and vice versa.

Algorithm 1 shows the working of the external event extraction from the annotated Twitter data by analysing the semantic, spatial, and temporal characteristics of the tweets. It takes the annotated tweets reported in a specified time interval Δt and then clusters each tweet, represented by the tuple (e, t, l), to potential external event. First of all, the semantic similarity of event terms *e* with the relevant external event *EE* is computed (lines 5–10). If the similarity between *e* and *EE* is above β, then the meta data of that tweet is linked to *EE* (line 8). All event terms *e* which have been assigned the same *EE* are grouped into one event type represented as *E[type]* (lines 12–14). The grouped clusters are then filtered based on location to match the *locbdy* of *EE* (lines 15–18). We assume that most of the tweets discussing a particular external event *EE* should be located within the proximity of its defined *locbdy* in order to

```
Algorithm 1: External Event Extraction Algorithm
Data: $(e, t, l)$ = Tweet contents as input feature vector
      $N$ = Total number of tweets
      $\Delta t$ = Time step such as hour, day, or week used to step through starting time $t_s$ and ending time $t_e$
      $\beta$ = Similarity threshold value
Result: $EV_n$ = Event tuple where $n$ represents the number of potential relevant external events
1. while $\exists\, (e, t, l)$ within a time interval $\Delta t$ do
2.    for $i = 1{:}N$ do
3.      $v_i = (e, t, l)_i$ ;
4.      Compute semantic similarity of event term $e$ with relevant external event $EE$;
5.      for $i = 1{:}k$ do
6.         if ( SemanticSimilarity(e, $EE_k$) > $\beta$)
7.           then
8.           $v_i = EE_k$ ; // external event assignment to event term $v_i$
9.         end if
10.     end for
11.   end for
12.   for $i = 1{:}N$ do
13.     Collect all the feature vectors $v_i$ with the same type into an event type list $E[type_k]$ where $k$ is the number of event types;
14.   end for
15.   for $i = 1{:}k$ do
16.     $loc_g$ = Find all grid locations with tweets higher than threshold in $E[type_i]$ ;
17.     foreach $l_g$ in $loc_g$
18.        if(match($locbdy$, $l_g$))
19.          $Pr(E[type_i]) = AvgSim(E[type_i]) * (\text{strength of } E[type_i] \text{ at } l_g / \text{Total strength of } E[type_i])$;
20.        end if
21.     end for
22.   end for
23.   for $i = 1{:}k$ do
24.     $e_{i,type} = E[type_i]$;
25.     $e_{i,loc} = l_g$ that matched to $locbdy$ of $E[type_i]$;
26.     $e_{i,st}$ = smallest timestamp in the cluster $E[type_i]$ ;
27.     $e_{i,et}$ = largest timestamp in the cluster $E[type_i]$ ;
28.     $e_{i,pr} = Pr(E[type_i])$;
29.     $EV = (e_{i,type}, e_{i,loc}, e_{i,st}, e_{i,et}, e_{i,pr})$;
30.   end for
31. end while
```

have any impact on the SLA. Therefore, the location of tweets clustered in *E[type]* are matched with the *locbdy* of that external event *EE* (line 18). The uncertainty in determining the content and location of the relevant external event is represented by the *Pr(E[type])* (line 19). If the majority of tweets belonging to *E[type]* are within *locbdy*, then the probability of *EE*'s occurrence is higher and vice versa. Finally, all the processed *E[type]*s are returned in the form of a vector *EV* (line 23–30), where each *E[type]* is identified by e_{type}, as shown in Eq. (3).

$$EV = (e_{type}, e_{loc}, e_{st}, e_{et}, e_{pr}) \tag{3}$$

where e_{type} represents relevant external event,

e_{loc} represents its location,

e_{st} and e_{et} represents its starting time and ending time respectively, and

e_{pr} represents its probability.

From this stage onwards, we only consider e_{type} as the relevant external events rather than EE, because from Eq. (2), not all EEs will satisfy the required threshold of Ω for them to be considered any further.

6. Impact assessment module

The aim of this module is to ascertain the impact of each relevant external event e_{type} on the GT to which it is linked in Fig. 4. In real-world CoT applications, the consequences of an external event e_{type} impacting a GT can be experienced in many different business KPIs, such as financial, service, reputation etc. In this paper, we focus on financial and service impacts. Financial impact refers to the monetary loss incurred due to disruption and is represented as *financial loss* (f_{loss}) whereas, service impact refers to the failure of a service providing the required QoS due to disruptions [5] and is represented as *recovery time* (r_{time}) as given in Table 3. Such impacts on a GT will have a snowball effect on other GTs that are dependent on it. As our objective in this paper is to pre-ascertain the chances of SLA violation occurring, we focus on the service QoS factors only while determining the impact which the external event will have on the GTs.

To ascertain the impact of e_{type} on GT, we first need to ascertain the *severity of disruption* (s_{disrup}) caused by it. As shown in Eq. (4), s_{disrup} is dependent on two factors, namely the *recovery time* (r_{time}) and *financial loss* (f_{loss}).

$$sdisrup_{etype} = rtime_{etype} \odot floss_{etype} \tag{4}$$

where $sdisrup_{etype} \in (0, 1)$ represents the severity of disruption on the GT due to external event e_{type}, $rtime_{etype}$ represents the time required for the CoT SLA to recover in its operations due to e_{type}, $floss_{etype}$ represents the financial loss that will be incurred due to e_{type}, and $\odot$ is a fuzzy Mamdani join operator.

Table 3
History of past disruptions due to external events along with the recovery time and financial loss [5].

Disruption cause	Description (e_{type})	Discovery	Recovery time (r_{time})	Financial loss (f_{loss})
External	Terrorism	Same day	1 week	Tens of millions
External	Supplier labor strike	Same day	100 days	Unable to disclose
Natural disaster	Snowstorm	Same day	Weeks	Expired product, unknown
External	Terrorism	Same day	weeks	Cost not important
External	3PL merger	2–3 weeks	2–4 months	$0, cost recovered from provider
Natural disaster	Hurricane	Same day	Unknown	–
Internal	Product quality failure	Weeks	Months	$100 million
Natural disaster	Flooding	Same day	unknown	Some freight costs
Internal	Product recall	weeks	6 months	$50-60 millions
External	Supplier quality	Same day	45 days	$0, cost recovered from supplier
External	War in supplier's country	1 week	weeks	Cost not important
External	Supplier quality	Same day	weeks	20%–30% premium transportation
Internal	Improper import docs.	1 week	2 weeks	"considerable payments to customer"
External	Supplier delay (transport)	Same day	2 weeks	>$60,000
External	Supplier quality	Weeks	10 months	Air freight costs
External	Supply chain coordination	Weeks	3 months	Overtime labor costs
Internal	Product recall	Same day	Unknown	$5 million
External	Transportation labor strike	Same day	2-4 weeks	>$300,000
External	Supplier quality	Weeks	Weeks	Unknown
External	Raw material shortage	Weeks	Months	$1 million
External	3PL warehouse startup	Same day	8 months	$3-7 million
Natural disaster	Hurricane	Same day	Unknown	Small financial impact
External	Customs delay	Same day	Unknown	Premium freight
External	Customs delay	Same day	1-2 weeks	<$100,000
Natural disaster	Snowstorm	1 week	<2 months	$0
Natural disaster	Hurricane	Same day	2 weeks	$0
External	Customs delay	Same day	6 days	$0
External	Product transport damage	Same day	2 days	Minor labor costs
Internal	Product quality failure	Same day	Weeks	$2 million
Natural disaster	Hurricane	Same day	Weeks	$10 million
External	Demand spike	1 week	12 weeks	Minor freight costs
External	Customs delay	Same day	4 months	$0
External	Transportation availability	Same day	10 weeks	$0
External	Supplier quality	Same day	Unknown	$0

Table 4
Fuzzy rules for measuring the *severity of disruption* ($sdisrup_{etype}$) due to external event e_{type}.

If	Recovery time (r_{time})	and	Financial cost (f_{loss})	then	$sdisrup_{etype}$
If	High	and	High	then	High
If	High	and	Medium	then	High
If	High	and	Low	then	High
If	Medium	and	High	then	High
If	Medium	and	Medium	then	High
If	Medium	and	Low	then	Low
If	Low	and	High	then	High
If	Low	and	Medium	then	Low
If	Low	and	Low	then	Low

To determine the values for r_{time} and f_{loss}, we consider that the CoT SLA manager will have the information shown in Table 3 [5]. The table represents the past impacts which were experienced due to external event e_{type}, the time required to recover from them (r_{time} - represented by the column recovery time of Table 3) and the financial loss incurred due to that event (f_{loss} is represented by the column financial loss). A Mamdani fuzzy inference system is used by the CoT SLA manager to translate the linguistic values for r_{time} and f_{loss} to fuzzy sets of *low, medium* and *high* and the resulting $sdisrup_{etype}$ is determined on the fuzzy sets of *low* or *high* by using fuzzy rules of Table 4.

After this, the impact of the external event e_{type} on a GT can be determined by using Eq. (5).

$$Impe_{type\rightarrow GT} = e_{pr} \oplus sdisrup_{e_{type\rightarrow GT}} \tag{5}$$

where $Impe_{type\rightarrow GT} \in (0, 1)$ represents the impact of the external event e_{type} on a GT, e_{pr} comes from Eq. (3), which represents the probability of occurrence of the external event e_{type}. $sdisrup_{e_{type\rightarrow GT}}$ from Eq. (4) represents the severity of disruption on the GT due to the external event e_{type}. $\oplus$ is a fuzzy Mamdani join operator.

As $sdisrup_{etype\rightarrow GT}$ is determined on the fuzzy sets, $Impe_{type\rightarrow GT}$ too is determined as a fuzzy variable before being defuzzified to ascertain its crisp output. Table 5 presents the fuzzy rules for determining $Impe_{type\rightarrow GT}$ on the fuzzy sets of low and high. The membership function of $Impe_{type\rightarrow GT}$ is shown in Fig. 8. By defuzzifying the fired rules of Table 5, a crisp value for $Impe_{type\rightarrow GT}$ value over a range of (0, 1) inclusive is obtained.

Table 5
Fuzzy rules for impact of e_{type} on a GT ($Impe_{type\rightarrow GT}$).

If	e_{pr}	and	$sdisrup_{etype\rightarrow GT}$	then	$Impe_{type\rightarrow GT}$
If	High	and	High	then	High
If	High	and	Low	then	Low
If	Low	and	High	then	High
If	Low	and	Low	then	Low

In the scenario of more than one relevant external event e_{type} impacting a GT (for example GT_1 of Fig. 12) then the $Impe_{type\rightarrow GT}$ is computed as shown in Eq. (6) bounded to a maximum value of 1.

$$Impe_{type\rightarrow GT} = \sum_{i=1}^{n} Impe_{typei\rightarrow GT} \tag{6}$$

$$\mu_{Impe_{type\rightarrow GT\ Low}}(x) = \begin{cases} 0, & x = 1 \\ \dfrac{1-x}{1}, & 0 < x < 1 \\ 1, & x = 0 \end{cases}$$

$$\mu_{Impe_{type\rightarrow GT\ High}}(x) = \begin{cases} 1, & x = 1 \\ \dfrac{x-1}{1}, & 0 < x < 1 \\ 0, & x = 0 \end{cases}$$

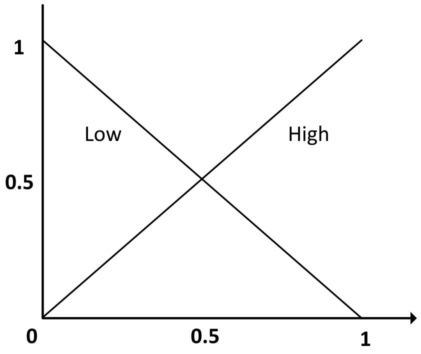


Fig. 8. Fuzzy membership function for $Impe_{type\rightarrow GT}$.

where $Impe_{type\rightarrow GT}$ represents the impact of a relevant external event e_{type} on a GT, n represents the number of relevant external event e_{type} impacting a GT.

7. Reasoning and decision-making module

The objective of this module is to ascertain if the SLA's state is going to be *violated* or not. It will do this by utilizing the information captured until now which is related to the SLA, the compositor relationships between its GTs and the impact of the relevant external events on them. If the SLA's state is determined as violated, then the CoT SLA manager can immediately take preventive steps to bring the SLA's state back to *satisfied* and avoid the consequences. To achieve the intended goal, the *reasoning and decision-making* module has two steps. The first step initializes the reasoning and decision-making module in two aspects. The first aspect takes the nested relationships present between SLA GTs as shown in Fig. 3 and the second aspect models it by using the compositor relationships present between a CGT and its constituting GTs (from Section 4.3). This is utilized in the second step, where based on a set of observed events e which can be both internal and external events, reasoning is performed to ascertain the final state of the SLA. The working of each step is explained in the next sections.

7.1. *Initializing the reasoning and decision-making module*

Event calculus as an approach is used to model the nested and composite relationship present between the GTs [4]. Algorithm 2 shows the working of this module. An initial knowledge base ($\mathrm{KB}_{(0)}$) with initial facts, domain-independent axioms (Σ) and domain axioms ($\Delta_{(0)}$) is initialized (line 1–3). The QoS constraints of GTs defined in the SLA are represented as *state constraints* (Ψ) (line 4). Next, *effect constraints* (E_1) that represent the transition of GTs from one state to another according to the composite relationships present between them are defined (line 5). These dependencies are captured in the form of rules and stored in the KB. When a set of observed events e (from Section 7.2) is given to the reasoning and decision-making module, then some of these rules represented by the symbol $\bar{O}_{(t)}$ (line 7) are triggered to ascertain the final state of a GT.

7.2. *Reasoning process based on a set of observed events e*

The reasoning process takes as inputs the sequence of observed events e and processes them to determine the GTs state at time $t+1$, represented by $\mathrm{KB}_{(t+1).}$ The updates are validated against the state constraints to determine the states of a GT at time $t+1$ (lines 8–10). At that time, a GT can either be a *known* or *unknown* GT.

(i) *Known G:T* Those GTs whose states at a time instant $t+1$ are known either as a direct result of the impact from observed events e or due to its interdependence on/from other GTs are termed as *known GTs*. In other words, known GTs are those GTs $O = \{gt_1, \mathrm{gt}_2, \ldots, gt_n\}$ whose state at any time $t+1$ is known either to be in a *violated, satisfied* or *uncertain* state due to the occurrence of observed events e (line 11–13)

(ii) *Unknown GT:* Those GTs whose states at a time instant $t+1$ are not known are termed as *unknown* GTs. In other words, unknown GTs are those GTs whose current state cannot be determined when the observed events e impact the SLA and hence their future states are set as not known.

Once the set of observed events e are processed, then the final state of the SLA at time $t+1$ is ascertained to see if it is *violated* or not. If it is in a violated state, then the reasoning process stops as the CoT SLA manager needs to immediately take preventive steps to bring the SLA's state back to *satisfied*. It is important to note here that it is not necessary to know the state of all the GTs to ascertain the final state of the SLA. For instance, from Fig. 9 it can be seen that the states of GT2, GT_3 and GT_5 are known at any time instant t. However, at the same time, the states of GT1 and GT4 are unknown which hinders the determination of the final state of the SLA at time point $t+1$. But if any of the known GTs (i.e. GT_2 or GT_3) are in a *violated* state, then we know that the SLA is in a *violated* state as the compositor between the SLA and GTs is of type *All*. In another scenario, if neither GT_2 nor GT_3 are in a *violated* state, then the state of GT_1 needs to be determined to ascertain the final state of the SLA. If the composite join between GT_4 and GT_5 is *OneOrMore or ExactlyOne*, then we need to know if the state of the *unknown* GT_4 is either *satisfied or violated* or *uncertain* before being able to ascertain the final state of the SLA. This is achieved by probabilistic inferencing using Bayesian networks (line 16), as explained in the next sub-section. When the state of these GTs is known, the knowledge base is updated with the new information to determine the future state of the SLA (lines 17–20).

7.3. *Probabilistic inferencing using bayesian networks to ascertain the state of unknown GTs*

When there is interdependency between the known and unknown GTs as shown in Fig. 10(b), then in such situations, the state of the unknown GTs can be determined by using dependency relationships. Bayesian networks provide full representations of probability distributions and can be conditioned upon any subset of their variables, supporting any direction of reasoning [43]. The direction of reasoning can be between nodes to ascertain the causes of an effect. To determine the state of unknown GTs, we use a Bayesian network (BN) which is a probabilistic model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. The formed BN model

```
Algorithm 2: Proactive Reasoning Algorithm
Data: QoS observations at timeslot t
Result: updated $KB_{(t+1)}$
Initialization;
initialize $KB_{(0)}$ with domain axioms $\Delta_{(0)}$ and initial facts;
initialize EC domain-independent axioms $\Sigma$
activate EC state constraints $\Psi$
activate EC effect constraints $E_1$
loop
    input QoS observations at timeslot t and update observation axioms $\bar{O}_{(t)}$
    for all $KB_{(t)}$ at timeslot t do
        $\acute{K}\beta_{(t+1)} \leftarrow KB_{(t)} \cup \bar{O}(t) \cup \Sigma$
        $\acute{K}\beta_{(t+1)} \leftarrow \acute{K}\beta_{(t+1)} \cup E_1$
        if $\acute{K}\beta_{(t+1)}$ is consistent with $\Psi$ then
            $KB_{(t+1)} \leftarrow \acute{K}\beta_{(t+1)} \cup \Psi$
            store $KB_{(t+1)}$
        else
            $\bar{Q}_{(t)} \leftarrow$ produce all combinations of truth values for the fluents which are not affected at timeslot t
            $\bar{Q}$ Set $\leftarrow$ Perform Probabilistic Inference($\bar{Q}_{(t)}$)
            foreach rs $\in \bar{Q}$ Set do
                if $\acute{K}\beta_{(t+1)} \cup$ rs is consistent with $\Psi$ then
                    $KB_{(t+1)} \leftarrow \acute{K}\beta_{(t+1)} \cup \Psi \cup$ rs
                    store $KB_{(t+1)}$
                end if
            end for
        end if
    end for
end loop
```

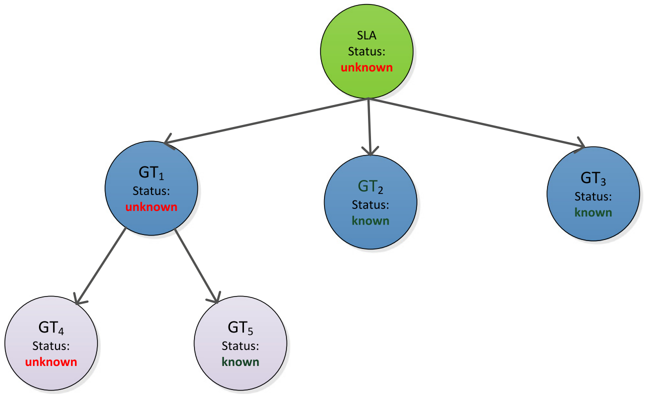


Fig. 9. The change in the SLA's final state according to the compositor relationship between the GTs.

can be of one of two types, as shown in Fig. 10. Fig. 10(a) illustrates the scenario when there is no relevant external event impacting the GT, whereas Fig. 10(b) illustrates the scenario of the impact of a relevant external event on the GT. The algorithm 2 determines the state of GTs for both scenarios, which are also discussed in detail in the following sub-sections.

7.3.1. Ascertaining the state of a GT when no relevant external event impacts on it

Let us assume that at a specific time instant '*t*', we know the state of GT_1 from Fig. 10(a) while the state of GT_2 is unknown. Then, according to the Bayesian rule, the probability of CGT being in a violated state given the state of GT_1 as evidence is:

$$Pr\ (CGT|GT_1) = \frac{\sum_{GT_2 \in s,v,u} Pr\ (CGT)\ Pr(GT_1, GT_2|CGT)}{\sum_{GT_2 \in s,v,u} Pr(GT_1, GT_2)} \tag{7}$$

In Eq. (7) above, the expression $Pr(GT_1, GT_2|CGT)$ is difficult to estimate when the number of terms increases and therefore, it is normally replaced with $Pr\ (GT1|CGT)\ Pr(GT2|CGT)$ as all the GT_s are independent [44]. Therefore, Eq. (7) takes the following form:

$$Pr\ (CGT|GT_1) = \frac{\sum_{GT_2 \in s,v,u} Pr\ (CGT)\ Pr\ (GT_1|CGT)\ Pr(GT_2|CGT)}{\sum_{GT_2 \in s,v,u} Pr(GT_1, GT_2)} \tag{8}$$

From Eq. (8) above, the term $\sum_{GT_2 \in s,v,u} Pr(GT_1, GT_2)$ can be replaced by a normalization constant $\alpha = 1/\sum_{GT_2 \in s,v,u} Pr(GT_1, GT_2)$ [44]. Hence, Eq. (8) can be written as:

$$Pr\ (CGT|GT_1) = \alpha \sum_{GT_2 \in s,v,u} Pr\ (CGT)\ Pr\ (GT_1|CGT)\ Pr(GT_2|CGT) \tag{9}$$

Referring back to the scenario depicted in Fig. 10(a), if we know the states for GT_1 is violated which is represented as v_1, then we can find the probability of the *CGT* being in a *violated* state when GT_2 is unknown using Eq. (9):

$$\begin{aligned} &Pr\ (CGT\ (violated)\ |v_1) \\ &= \alpha (Pr\ (violated)\ Pr\ (v_1|violated)\ Pr\ (s_2|violated) \\ &+ Pr\ (violated)\ Pr\ (v_1|violated)\ Pr\ (v_2|violated) \\ &+ Pr\ (violated)\ Pr\ (v_1|violated)\ Pr\ (u_2|violated)\ ) \end{aligned} \tag{10}$$

7.3.2. Ascertaining the state of a gt while considering the impact of a relevant external event

As shown in Fig. 10(b), in this scenario, the composite term (CGT) has two guarantee terms GT_1 and GT_2 while GT_1 is affected due to a relevant external event e_{type}. Let us suppose we know the state of GT_1 and impact of e_{type} on it while the state of GT_2 is unknown. Then, according to the Bayesian rule, the probability of CGT being in a certain state given the state of GT_1 and e_{type} as

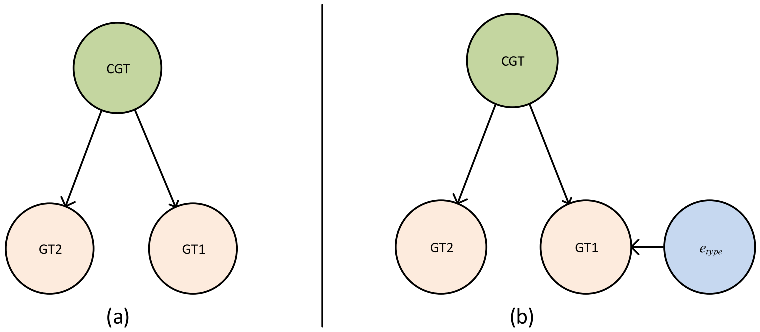


Fig. 10. Dependency relationships between the GTs and relevant external event.

evidence is:

$$Pr\left(CGT|GT_1 \wedge e_{type}\right) = \frac{\sum_{GT_2\in s,v,u} Pr\ (CGT)\ \ Pr(GT_1, GT_2, e_{type}|GT) Impe_{type\to GT}(e_{type}|GT_1)}{\sum_{GT_2\in\{s,v,u\}} Pr\left(GT_1, GT_2, e_{type}\right)} \tag{11}$$

Similar to Eqs. (8) and (9), Eq. (11) takes the following form:

$$Pr\left(CGT|GT_1 \wedge e_{type}\right) = \alpha \sum_{GT_2\in s,v,u} Pr\ (CGT) \times Pr\ (GT_1|CGT)\ Pr(GT_2|CGT) Impe_{type\to GT}(e_{type}|GT_1) \tag{12}$$

Referring to the scenario depicted in Fig. 10(b), if we know that the state of GT_1 is violated which is represented as v_1, the impact of the relevant external event e_{type} on GT_1 ($Impe_{type\to GT}(e_{type}|GT_1)$) as *high* and the probability of e_{type} being *high* as evidence, then we can find the probability of *CGT* being in a *violated* state when GT_2 is unknown

$$\begin{aligned} Pr\ (CGT(violated)|v_1 \wedge high) &= \alpha(Pr\ (violated)\ Pr\ (v_1|violated) \\ &\times Pr\ (s_2|violated)\ Pr\ (high|v_1) \\ &+ Pr\ (violated)\ Pr\ (v_1|violated)\ Pr\ (v_2|violated)\ Pr\ (high|v_1) \\ &+ Pr\ (violated)\ Pr\ (v_1|violated)\ Pr\ (u_2|violated)\ Pr\ (high|v_1)\ ) \end{aligned} \tag{13}$$

Moreover, in this case, the probability of GT_1 can also be computed first using the evidence of the relevant external event, which can then be used to determine the state of CGT. For example, if $Pr\ (GT_1)$ represents the probability of GT_1 and $Impe_{type\to GT}(e_{type}|GT_1)$ represents the impact of e_{type} on GT_1. Then $Pr\left(GT_1|e_{type}\right)$ can be computed as follows:

$$Pr\left(GT_1|e_{type}\right) = \frac{Pr\ (GT_1)\ Impe_{type\to GT}(e_{type}|GT_1)}{Pr\left(e_{type}\right)} \tag{14}$$

This gives the probability of GT_1 after considering the evidence of the impact of relevant external event e_{type}. In this Eq. (14), $Pr\ (GT_1)$ describes the probability of GT_1 being in either a *satisfied*, *violated* or *uncertain* state. $Pr\left(e_{type}\right)$ comes from the probability computation of relevant external event (e.g. from Twitter streams), whereas $Impe_{type\to GT}(e_{type}|GT_1)$ comes from the impact assessment of the external event on the partner organization. This impact value is used to form a conditional probability table (CPT) for $Pr\left(e_{type}|GT_1\right)$. For example, the probability of GT_1 being violated (v_1) can be determined given the probability of relevant external event e_{type} being *high* (e) as follows:

$$\Pr\ (v_1|e) = \frac{Pr\ (v_1)\ Pr\ (e|v_1)}{Pr\ (e)} \tag{15}$$

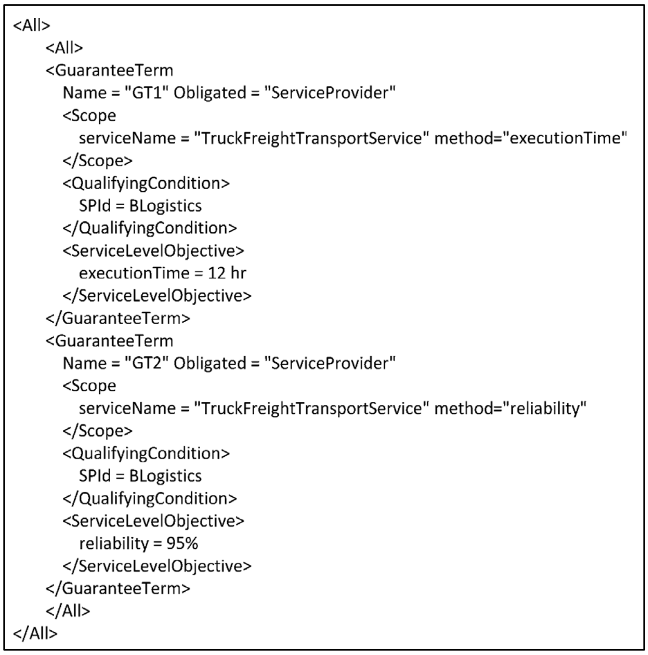
In the next section, we present the experimental validation of the proposed framework to show its superiority in proactively identifying SLA violations compared to the existing approaches that do not capture the external events.

8. Validation of the proposed framework for proactive SLA violation identification by capturing relevant external events

In this section, we assess the suitability of the proposed system for proactive SLA violation identification. We run simulations to identify SLA violations by considering the relevant external events and their impact on the GTs and compare the performance with the existing approaches that do not capture such external events. At the end, we ascertain the prediction quality by using the accuracy of violation identification to compare the results.

To model the SLA, we take a simplified example of a logistics application using a CoT environment as described in [45]. The application is of a logistics provider that transports goods in multiple cities such as Sydney, Canberra, Melbourne, Adelaide, Darwin and Perth using local delivery services. Each leg of the journey may be undertaken by a local transportation company servicing a local area. Let us suppose Company A (service user) uses the service of the aforementioned logistics provider for the transportation of its goods. Two requirements of Company A, apart from the costs, are execution (delivery) time and reliability of the service. Based on this, Company A signs an SLA with the logistics provider to ship goods from Sydney to Canberra. A segment of the formed SLA in the WS-Agreement is shown in Fig. 11 and a hierarchical nested structure of the GTs of this SLA is illustrated as *Model 1* in Fig. 12. It consists of two guarantee terms (GT_1 and GT_2) which are combined by an *All* compositor with the SLA. GT_1 specifies the execution time for sending goods from source to destination while GT_2 specifies the reliability of the logistics provider. As the compositor between the GTs is *All*, the SLA can be violated if any one of the two GTs is violated. Let us assume that Company A checks the previous record of the logistics company only based on internal events and forms an SLA with it. However, as mentioned in this paper and depicted in model 2 of Fig. 12, relevant external events arising from environmental factors impacting on the locations of interest impact on the partner organization's ability to adhere to the terms of a GT mentioned in the SLA [8]. We consider two such events in this example, namely inclement weather *rainstorm* on route (e_{type1}) and a *strike* by the customs officials (e_{type2}) which have an impact on execution time (GT_1) as shown in Model 2 of Fig. 12. Real-time information on such external events of interest can be found in the external sources (e.g. social media). We consider that the logistics company employs Algorithm 1 to identify information of the potential external events of interest impacting a GT and the impact assessment using the information in Table 3.

We created two models for SLA violation determination, as shown in Fig. 12. In model 1, we do not take into account the effect of relevant external events in violation prediction but we do consider the nested relationship between the GTs and SLA using



```
<All>
    <All>
    <GuaranteeTerm
      Name = "GT1" Obligated = "ServiceProvider"
      <Scope
        serviceName = "TruckFreightTransportService" method="executionTime"
      </Scope>
      <QualifyingCondition>
        SPId = BLogistics
      </QualifyingCondition>
      <ServiceLevelObjective>
        executionTime = 12 hr
      </ServiceLevelObjective>
    </GuaranteeTerm>
    <GuaranteeTerm
      Name = "GT2" Obligated = "ServiceProvider"
      <Scope
        serviceName = "TruckFreightTransportService" method="reliability"
      </Scope>
      <QualifyingCondition>
        SPId = BLogistics
      </QualifyingCondition>
      <ServiceLevelObjective>
        reliability = 95%
      </ServiceLevelObjective>
    </GuaranteeTerm>
    </All>
</All>
```

Fig. 11. Logistics SLA formed between Company A and Logistics provider.

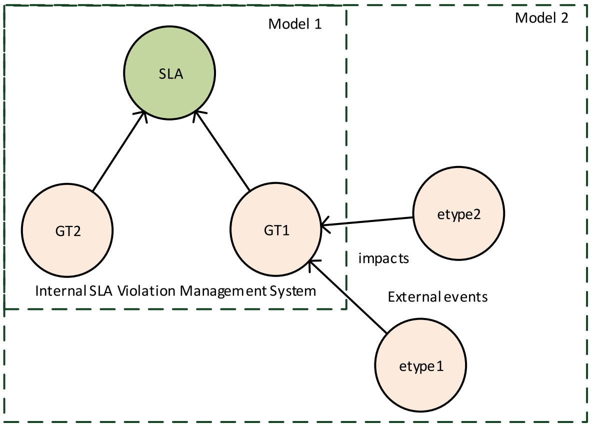


Fig. 12. Graphical Representation of Logistics SLA with and without the impact of relevant external events.

Table 6
Conditional probability table for GT_1 given GT.

GT	GT_1		
	S	**V**	**U**
Satisfied	0.934	0.016	0.049
Violated	0.017	0.593	0.390
Uncertain	0.091	0.091	0.818

Table 7
Conditional probability table for GT_2 given GT.

GT	GT_2		
	S	**V**	**U**
Satisfied	0.0164	0.770	0.213
Violated	0.017	0.797	0.186
Uncertain	0.091	0.091	0.818

Table 8
Probability for GT (SLA node).

Satisfied	Violated	Uncertain
0.472	0.456	0.072

Table 9
Conditional probability table for e_{type} and its impact on GT_1.

GT_1	e_{type}	
	Low	**High**
S	0.845	0.155
V	0.028	0.972
U	0.029	0.971

the approach proposed in [4]. In other words, in model 1 only the logical relationship of GTs defined in the SLA and past data of the guarantees and their interdependence between them is taken into account to perform logical and probabilistic reasoning. However, in model 2, apart from logical relationship between the GTs and past data of the GTs, the impact of relevant external event on the GT proposed in this paper is also taken into account. A set of observed 200 events e which are a combination of internal and external events are given as inputs and for probabilistic inferencing, the conditional probabilities are calculated for the aforementioned scenario, as given in Tables 6–9. The letters S, V, and U in these tables correspond to the conditional probabilities of execution time (GT_1) and reliability (GT_2) being in satisfied, violated, and uncertain states, whereas *satisfied*, *violated*, and *uncertain* correspond to the probabilities GT (i.e. SLA node in Fig. 12) being in those states, given the states of GT_1. Table 9 shows the combined impact of relevant external event e_{type} on GT_1 as conditional probability. A high probability value of e_{type} shows the relevant external event is identified with a high probability of occurrence and given that value, it represents the conditional probability of GT_1 being in state S, V or U. Similarly, a low probability value of e_{type} shows the relevant external event is identified with a low probability of occurrence and given that value, it represents the conditional probability of GT_1 being in states S, V or U.

Using this evidence, we determined the SLA violation prediction for both models as shown in Table 10. The timestamp column shows the timeslot from the set of 200 processed events e, the outputs of Models 1 and 2 are shown in the format (predicted SLA state, probability) whereas the Actual SLA state shows the observed state. It can be observed that the violation prediction results of model 2, which uses evidence of the impact of a relevant external event on a GT, outperformed model 1 which did not consider such impact and only relied on the internal events impacting the SLA. The red entries in Model 1 and 2 highlight those timestamps in which the output state was different from the one actually observed.

The confusion matrix for the conducted experiment (on model 1 and 2) is presented in Table 11. The table shows the predicted states (by model 1 and 2) and observed states (actual) for 200 events. The observed column shows the breakdown of the observed states between satisfied, violated and uncertain for the 200 events. The predicted column shows the comparison of the two models i.e. model 1 - predicted with logical relationship within GTs but without relevant external events and model 2 - predicted with logical relationship within GTs and relevant external events. As shown in the table, Model 1 predicted only 19 out of a total of 28 observed violations correctly whereas Model 2 predicted 26 out of the 28 observed violations correctly. The total false predictions i.e. the events which were predicted as violated or uncertain but were observed as satisfied or predicted as satisfied or uncertain but were actually violated) are also higher for Model 1. This shows

Table 10
SLA violation prediction with (model 2) and without (model 1) considering relevant external events impacting the GT.

Timestamp	Model 1 output in (predicted SLA state, probability) proposed in [4]	Model 2 output in (predicted SLA state, probability) proposed in this paper	Actual SLA state
	…	…	
7	violated 0.859	violated 0.97	violated
8	violated 0.859	violated 0.97	violated
9	satisfied + 0.981	violated 0.859	violated
10	satisfied + 0.981	satisfied + 0.981	violated
11	violated 0.859	violated 0.97	violated
12	satisfied + 0.981	violated 0.859	violated
13	violated 0.859	violated 0.97	violated
14	uncertain + 0.559	violated 0.878	violated
	…	…	
36	satisfied + 0.981	violated 0.859	violated
37	satisfied + 0.933	uncertain + 0.559	violated
38	uncertain + 0.559	violated 0.878	violated
39	satisfied + 0.981	violated 0.859	violated
40	satisfied + 0.981	violated 0.859	violated
	…	…	

Table 11
Confusion Matrix comparing the results of Model 1 and Model 2 with the observed states.

States	Observed	Model 1 proposed in [4]			Model 2 proposed in this paper		
		Satisfied	Violated	Uncertain	Satisfied	Violated	Uncertain
Satisfied	152	142	10	0	142	10	0
Violated	28	7	19	2	1	26	1
Uncertain	20	0	0	20	0	0	20

Table 12
Overall Accuracy.

Accuracy Measure	Prediction without dependencies between GTs	Model 1 proposed in [4]	Model 2 proposed in this paper
Precision	0.731	0.907	0.951
Recall	0.855	0.905	0.940
F-measure	0.788	0.906	0.943

the overall increase in accuracy of the results from Model 2 from Model 1. The overall accuracy quality indicators for the conducted experiment are shown in Table 12, which shows a precision of 0.907 for Model 1 as compared to 0.951 for Model 2. Model 1 has a recall of 0.905 as compared to 0.940 for Model 2. Similarly, the F-measure for the prediction model with *relevant external events* is 0.943 whereas the prediction model without *relevant external events* is 0.906. The accuracy measures are also compared with another approach which ascertains the future state of the SLA without using the internal dependencies within GTs as done in [4]. It can be seen that the accuracy results of that model are very poor. These quality measures indicate a very good overall prediction accuracy for the prediction model with *relevant external events* compared to the prediction model without *relevant external events* but considering the internal dependencies between events (Model 1) and the model that does not consider this.

The conclusion that can be drawn from this experiment is that identifying potential relevant external events impacting GTs of SLA assists a CoT SLA manager to better manage its agreed commitments with service users and avoid SLA violations.

9. Conclusion and future work

In this paper, we focused on the problem of a CoT SLA manager considering the external events of interest to the formed SLAs for proactive management. This problem is of importance as services in CoT environments are formed across dispersed geographic locations. Hence, apart from only focusing on the usual internal events that may impact on the GTs, it is also necessary to consider the external events that will have an indirect impact on them, if an informed determination for SLA violation abatement is needed. From this perspective, in this paper, we explained our proposed framework along with its various inter-related modules. The major contributions of our proposed framework in the paper are: (a) it extends our previous work of considering the nested and hierarchical relationships between the GTs of SLAs to also consider the external events that may impact on it (b) it proposes an approach that utilizes Twitter as the source to extract information relevant to the external events (c) it ascertains the impact of the occurrence of relevant external events on the GTs and (d) it performs logical reasoning to proactively determine if a SLA is going to be violated or not. Using experiments, we demonstrated how the proposed approach will assist the CoT SLA manager in the better management of SLAs and how it is vastly different from the existing reactive approaches in the literature. In future work, we will extend our model to capture information relevant to external events from different sources such as news feeds apart from just Twitter. Another extension will be to identify the inter-dependencies between relevant external events and utilize these to identify SLA violations.

Acknowledgement

First author acknowledges UNSW for supporting his Ph.D. candidature with scholarship.

References

[1] S. Chun, S. Seo, B. Oh, K.H. Lee, Semantic description, discovery and integration for the Internet of Things, in: Proc. 2015 IEEE 9th Int. Conf. Semant. Comput. IEEE ICSC 2015, 2015, pp. 272–275.

[2] Y.-Y. Fanjiang, Y. Syu, J.-Y. Kuo, Search based approach to forecasting QoS attributes of web services using genetic programming, Inf. Softw. Technol. 80 (2016) 158–174.

[3] B. Cavallo, M. Di Penta, G. Canfora, An empirical comparison of methods to support QoS-aware service selection, in: Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems, 2010, pp. 64–70.

[4] F. Nawaz, N.K. Janjua, O.K. Hussain, F.K. Hussain, E. Chang, M. Saberi, Event-driven approach for predictive and proactive management of SLA violations in the Cloud of Things, Futur. Gener. Comput. Syst. 84 (2018) 78–97.

[5] J.R. Macdonald, T.M. Corsi, Supply chain disruption management: Severe events, recovery, and performance, J. Bus. Logist. 34 (4) (2013) 270–288.

[6] F. Aqlan, S.S. Lam, A fuzzy-based integrated framework for supply chain risk assessment, Int. J. Prod. Econ. 161 (2015) 54–63.

[7] Y. Sheffi, The Resilient Enterprise: Overcoming Vulnerability for Competitive Advantage, MIT Press, Cambridge MA, 2005.

[8] C. Blome, T. Schoenherr, D. Eckstein, The impact of knowledge transfer and complexity on supply chain flexibility: A knowledge-based view, Int. J. Prod. Econ. 147 (2014) 307–316.

[9] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: Real-time event detection by social sensors, in: Proc. 19th Int. Conf. World Wide Web, 2010, pp. 851–860.

[10] M.S. Gerber, Predicting crime using Twitter and kernel density estimation, Decis. Support Syst. 61 (1) (2014) 115–125.

[11] P. Anantharam, P. Barnaghi, K. Thirunarayan, A. Sheth, Extracting city traffic events from social streams, ACM Trans. Intell. Syst. Technol. 6 (4) (2015) 1–27.

[12] N. Alsaedi, P. Burnap, O. Rana, Can we predict a riot? Disruptive event detection using twitter, ACM Trans. Internet Technol. 17 (2) (2017) 1–26.

[13] W. K. B, J. Chen, J. Li, J. Liu, L. Liu, G. Osborne, N. Lothian, B. Cooper, T. Moschou, G. Neale, Carbon: forecasting civil unrest events by monitoring news and social media, in: Advanced Data Mining and Applications, in: Lecture Notes in Computer Science, vol. 10604, Springer, Cham, 2017, pp. 859–865.

[14] L. Zeng, C. Lingenfelder, H. Lei, H. Chang, Event-driven quality of service prediction, in: Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), in: LNCS, vol. 5364, 2008, pp. 147–161.

[15] P. Leitner, J. Ferner, W. Hummer, S. Dustdar, Data-driven and automated prediction of service level agreement violations in service compositions, Distrib. Parallel Databases 31 (3) (2013) 447–470.

[16] B. Tang, M. Tang, Bayesian model-based prediction of service level agreement violations for cloud services, in: Proc. - 2014 Int. Symp. Theor. Asp. Softw. Eng. TASE 2014, 2014, pp. 170–176.

[17] N. Wagner, Time series forecasting for non-static environments: the dyfor genetic program model, IEEE Trans. Evol. Comput. 11 (4) (2007) 433–452.

[18] F. Nawaz, M.R. Asadabadi, N.K. Janjua, O.K. Hussain, E. Chang, M. Saberi, An MCDM method for cloud service selection using a Markov chain and the best-worst method, Knowl.-Based Syst. 159 (Nov'17) (2018) 120–131.

[19] Y. Hu, Q. Peng, X. Hu, R. Yang, Web service recommendation based on improved collaborative filtering, IEEE Int. Conf. Web Serv. 8 (5) (2015) 782–794.

[20] O.K. Hussain, Zia-ur Rahman, F.K. Hussain, J. Singh, N.K. Janjua, E. Chang, A user-based early warning service management framework in cloud computing, Comput. J. 58 (3) (2015) 472–496.

[21] K. Boukadi, R. Grati, H. Ben-Abdallah, Toward the automation of a QoS-driven SLA establishment in the cloud, Serv. Oriented Comput. Appl. 10 (3) (2016) 279–302.

[22] M. Sánchez, J. Aguilar, E. Exposito, Fog computing for the integration of agents and web services in an autonomic reflexive middleware, Serv. Oriented Comput. Appl. (2018).

[23] S. Banerjee, M. Adhikari, U. Biswas, Design and analysis of an efficient QoS improvement policy in cloud computing, Serv. Oriented Comput. Appl. 11 (1) (2017) 65–73.

[24] F. Messina, G. Pappalardo, C. Santoro, D. Rosaci, G.M.L. Sarné, A multi-agent protocol for service level agreement negotiation in cloud federations, Int. J. Grid Util. Comput. 7 (2) (2016) 101–112.

[25] U. Villano, M. Rak, J. Modic, A. De Benedictis, V. Casola, Per-service security SLAs for cloud security management: model and implementation, Int. J. Grid Util. Comput. 9 (2) (2018) 128.

[26] S. Bitam, A. Mellouk, S. Zeadally, VANET-cloud: A generic cloud computing model for vehicular Ad Hoc networks, IEEE Wirel. Commun. 22 (1) (2015) 96–102.

[27] E. Ben Hamida, M.A. Javed, W. Znaidi, Adaptive security provisioning for vehicular safety applications, Int. J. Space-Based Situated Comput. 7 (1) (2017) 16–31.

[28] H. Choi, H. Varian, Predicting the present with google trends, Econ. Rec. 88 (Suppl. 1) (2012) 2–9.

[29] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, J. Comput. Sci. 2 (1) (2011) 1–8.

[30] U. Kursuncu, M. Gaur, U. Lokala, K. Thirunarayan, A. Sheth, I.B. Arpinar, Predictive analysis on twitter: techniques and applications, Emerg. Res. Challenges Oppor. Comput. Soc. Netw. Anal. Min. (2018) 1–37.

[31] M. Ebrahimi, A.H. Yazdavar, A. Sheth, Challenges of sentiment analysis for dynamic events, IEEE Intell. Syst. 32 (5) (2017) 70–75.

[32] A. Sheth, A. Jadhav, P. Kapanipathi, C. Lu, H. Purohit, G.A. Smith, W. Wang, Twitris: a system for collective social intelligence, Encycl. Soc. Netw. Anal. Min. (2014) 2240–2253.

[33] C.A. Davis, G.L. Ciampaglia, L.M. Aiello, K. Chung, M.D. Conover, E. Ferrara, A. Flammini, G.C. Fox, X. Gao, B. Gonçalves, P.A. Grabowicz, K. Hong, P.-M. Hui, S. McCaulay, K. McKelvey, M.R. Meiss, S. Patil, C. Peli Kankanamalage, V. Pentchev, J. Qiu, J. Ratkiewicz, A. Rudnick, B. Serrette, P. Shiralkar, O. Varol, L. Weng, T.-L. Wu, A.J. Younge, F. Menczer, OSoMe: the IUNI observatory on social media, PeerJ Comput. Sci. 2 (2016) e87.

[34] D.E. O'Leary, The use of social media in the supply chain: survey and extensions, Intell. Syst. Account. Financ. Manag. 16 (1–2) (2009) 21–31.

[35] B. Chae, Insights from hashtag #supplychain and Twitter analytics: Considering Twitter and Twitter data for supply chain practice and research, Int. J. Prod. Econ. 165 (2015) 247–259.

[36] T.N.A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, M.X.J. Pruyne, J. Rofrano, S. Tuecke, Web services agreement specification (WS-agreement), 2007.

[37] Y. Xu, J. Yin, S. Deng, N.N. Xiong, J. Huang, Context-aware QoS prediction for web service recommendation and selection, Expert Syst. Appl. 53 (2016) 75–86.

[38] L.A. Ramshaw, M.P. Marcus, Text chunking using transformation-based learning, 1995, pp. 157–158.

[39] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015, in arXiv:1508.01991v1 [cs.CL].

[40] M.A. Mirtalaie, O.K. Hussain, E. Chang, F.K. Hussain, Extracting sentiment knowledge from pros/cons product reviews discovering features along with the polarity strength of their associated opinions, Expert Syst. Appl. 114 (2018) 267–288.

[41] F. Benamara, S. Irit, C. Cesarano, N. Federico, D. Reforgiato, Sentiment analysis: Adjectives and adverbs are better than adjectives alone, in: Proc Int Conf Weblogs Soc. Media, 2007, pp. 1–4.

[42] D. Sánchez, M. Batet, D. Isern, A. Valls, Ontology-based semantic similarity: A new feature-based approach, Expert Syst. Appl. 39 (9) (2012) 7718–7728.

[43] K.B. Korb, A.E. Nicholson, Bayesian Artificial Intelligence, second ed., CRC Press, Inc., Boca Raton, FL, USA, 2010.

[44] D. Heckerman, D.M. Chickering, D. Geiger, D.M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Mach. Learn. 20 (3) (1995) 197–243.

[45] Q. He, J. Yan, R. Kowalczyk, H. Jin, Y. Yang, Lifetime service level agreement management with autonomous agents for services provision, Inf. Sci. (NY) 179 (15) (2009) 2591–2605.

Falak Nawaz is currently a Ph.D. candidate at School of Business, University of New South Wales (UNSW), Canberra, Australia. His research focuses on Service Level Agreement (SLA), Service Management, and SLA Violation Prediction. His research interests also include Service Computing, Cloud Computing, and Internet of Things.

Omar Hussain is a senior lecturer at the University of New South Wales, Canberra. His research interests are in business intelligence, cloud computing and logistics informatics. In these areas, his research work focuses on utilizing decision making techniques for facilitating smart achievement of business outcomes. His research work has been published in various top international journals such as Information Systems, The Computer Journal, Knowledge Based Systems, Future Generation of Computer Systems etc. He has won awards and funding from competitive bodies such as the Australian Research Council for his research.

Farookh Khadeer Hussain is an Associate Professor in School of Software, University of Technology Sydney. He is an Associate Member of the Advanced Analytics Institute and a Core Member of the Centre for Artificial Intelligence. His key research interests are in trust-based computing, cloud of things, blockchains and machine learning. He has published widely in these areas in top journals such as FGCS, The Computer Journal, JCSS, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics etc.

Naeem Khalid Janjua is a Lecturer at the School of Science, Edith Cowan University, Perth. He is an Associate Editor for International Journal of Computer System Science and Engineering (IJCSSE) and International Journal of Intelligent systems (IJEIS). He has published an authored book, a book chapter, and various articles in international journals and refereed conference proceedings. His areas of active research are defeasible reasoning, argumentation, ontologies, data modelling, cloud computing, machine learning and data mining. He works actively in the domain of business intelligence and Web-based intelligent decision support systems.

Morteza Saberi is a Research Fellow at UNSW Canberra and has an outstanding research records and significant capabilities in the area of Business Intelligence, Data Mining and applied machine learning. He has published more than 140 papers in reputable academic journals and conference proceedings. His Google Scholar citations and h-index are 1400 and 18 respectively. He was a Lecturer at the Department of Industrial Engineering at University of Tafresh. He is also the recipient of the 2006–2012 Best Researcher of Young researcher Club, Islamic Azad University (Tafresh Branch). He is also the recipient of National Eminent Researcher Award among Young researcher Club, Islamic Azad University members.

Elizabeth Chang is Professor and Canberra Fellow at the UNSW at the Australian Defence Force Academy (ADFA). She has 30 years of work experience in both Academia and Industry. She has been a full Professor in IT, Software Engineering and Logistics Informatics for 14 years. She had been in senior positions in commercial corporations for 10 years, typically working on commercial grade large software development. Her key research strength is in large complex software development methodologies, requirement engineering, structure and unstructured database design and implementation, trust, security, risk and privacy. In the 2012 edition of MIS Quarterly vol. 36 issue. 4 Special Issues on Business Research, Professor Chang was listed fifth in the world for researchers in Business Intelligence.