A stream processing framework based on linked data for information collaborating of regional energy networks

***Abstract*—Coordinating of energy networks to form a city-level multi-dimensional integrated energy system becomes a new trend in Energy Internet. Stream processing is the key technology in the information collaborating. However, the information flows are hard to represent and construct for heterogeneous multi-source data. Meanwhile, data incompleteness and fixed processing in streams decrease the precision and flexibility of stream processing. To solve these problems, we propose a stream processing framework based on linked data for information collaboration among multiple energy networks. The framework provides a universal data representation for stream derived from linked data to model heterogeneous data, leverages semantics based information fusion and division to ensure the data completeness, and establishes exchanging channels and processing windows with variable granularity to achieve dynamic stream processing. A real-world case study is implemented to demonstrate the adaptability, feasibility and flexibility of the proposed framework.**

***Index Terms*—Linked Data, web semantics, data fusion, stream processing, Internet of Things.**

I. Introduction

City-level multi-dimensional energy network with deep integration of multiple energy forms is the key application in Energy Internet [1] to achieve multi-energy complementarity and demand response. Energy Internet is a typical case of Internet of Things(IoT) [2] where stream processing [3] plays an important role. Stream processing, which supports online data collection, computing and transmission, provides a continuous and efficient mechanism for system monitoring and management. In large cities, there are several isolated regional energy networks located in distributed space. To achieve a higher-level energy optimal utilization, development in energy hub [4] brings new potential in energy flow exchanging. And advancements in stream processing present the possibility of information sharing. However, to adopt stream processing in information collaborating among multiple energy networks, the main challenges lies in three aspects.

(1) Heterogeneous data from different energy networks are hard to dispose of in stream format. Each utility component in a network has its information model. These models are different in formats (such as key-values, files, SQL tables or rows, etc.) and semantics (such as 'incoming water' and 'inlet water'), so as the generated information flows such as water temperatures, pipe pressures, monitoring videos, etc. The heterogeneity in multi-source data makes the data collection more complex to unify them for interoperability in further processing. Therefore, a unified data description model becomes the first concern in stream processing for energy network information collaborating.

(2) Data incompleteness affects the accuracy of stream processing under volatile application scenes. Towards applications in a time slot, all the related data need to be prepared for reasonable analysis. For example, load, storage and pipe networks together influence the running of energy sources. A lack of data in each part leads to the wrong running command. But these data are collected from hundreds of thousands of machines with various updating frequencies, ie. spatially discrete and temporally inconsistent. It is hard to find all the data of the same type and machine of the same utility to avoid information loss. Therefore, it is necessary to discover the semantics and relations of data and automatically link related data to the analysis model.

(3) High volatility of energy IoT data requires variable granularity of stream processing in rapidly changing environments. Status of energy systems changes fast making online analysis inevitable. However, data are sent and used in different frequencies in different scenarios. Taking the energy source running control scenario as an example, the analysis time slot can be long when the energy load is small, but should be short when the load is near the existing supply ability. When several features collaborate, the adjustment of processing tempo is more complicated. Therefore, the dynamic adjustment to the granularity of stream processing is important.

To fulfill information flow directed deep integration of multiple energy and collaboration of multiple networks, we propose a stream processing framework based linked data [5] for unified data representation, automatic relation discovery, flexible exchanging control and versatile data processing. In this work, a universal data representation for IoT systems is designed as the foundation of interoperable digital twins for IoT systems. A dynamic information exchanging mechanism is proposed, which consists of semantic fusion directed exchanging contracts and channels, and provides a flexible way to connect adequate data towards applications. A data-driven time window adjustment approach is proposed to achieve the variable granularity of stream processing. Finally, a real-world case study is conducted on three regional energy networks demonstrate the adaptability, feasibility and flexibility of the proposed framework.

The rest of the paper is organized as follows. Section II introduces the related work in the field. Section III gives the complete landscape of the technology framework. Section IV presents the detailed process and methods. In Section V, a case study is shown and a discussion of the comparison with related work is illustrated. Finally, Section VI concludes the paper and the future directions of the work.

II. Related Work

For the real-time processing of Energy Internet, there are two main solutions to solve the latency challenge, the batch processing and the stream processing [3]. Considering the high volatility of data in Energy Internet, stream processing becomes the better choice. Flink [6] and Storm [7] are helpful tools as mature solutions. Dautov et al. [8] implement IoT/CPSS to process real-time data in distributed IoT systems. Puschmann et al. [9] applying stream clustering to IoT data stream for data fusion. However, how to self-organize and transmit data in complicated application scenes with variable granularity is still open.

To uniformly access the diverse data, many researchers have been working on different approaches. Semantic Sensor Network (SSN) ontology [10] is one of the widely-used ontology in the IoT domain. Graph of Things [11] is also a complete practice of combining stream processing, semantics and IoT. Chun et al. [12] design an IoT-DS which supports metadata updating, semantic description and discovery of objects in IoT. However, the domain-specific nature and complex rules in ontologies limit the ability of semantic extension, which lowers the adaptability of the approach. Then, applying Linked Data [5] as IoT data representation become an effective solution for large-scale data integration [13] for its scalability.

Addressing the data incompleteness issue, discovering semantic links [14] to improve is a mainstream direction. Milis et al. [15] propose SEMIoTICS to semantically model the components based on ontologies, and online compose and configure the control loops. Shi et al. [16] propose that exploiting linked open data to enrich the semantic information for better data fusion and knowledge discovery helps manage the information network. To intelligently manage the data communication, Bello et al. [17] conclude some intelligent device-to-device communication mechanism in IoT. The semantic relation assistant processing for information comprehensive communication still needs to be grounded.

The above researches respectively provide strong insights into the challenges of data acquisition, fusion, transmission and processing in complex IoT systems. Leveraging semantic data for unified data access is widely accepted in nowadays IoT systems. The ontology is one of important technology in semantic processing but is hard to extend. Linked data thus becomes a better choice. Although it is useful for data fusion and knowledge discovery, merely research has focused on applying it for information transmission. For stream processing, applying it to dynamically changed IoT systems is still an open issue. Therefore, a comprehensive data processing approach considering the scalability, flexibility and feasibility for IoT systems still remains to be revealed.

III. The Framework

To achieve the uniform data representation, dynamic information communication and real-time processing for multiple energy networks, a layered technology framework is illustrated in Fig. 1. It takes the metadata of subsystems as the input and goes through the four phases to achieve collaborative control of the whole system.

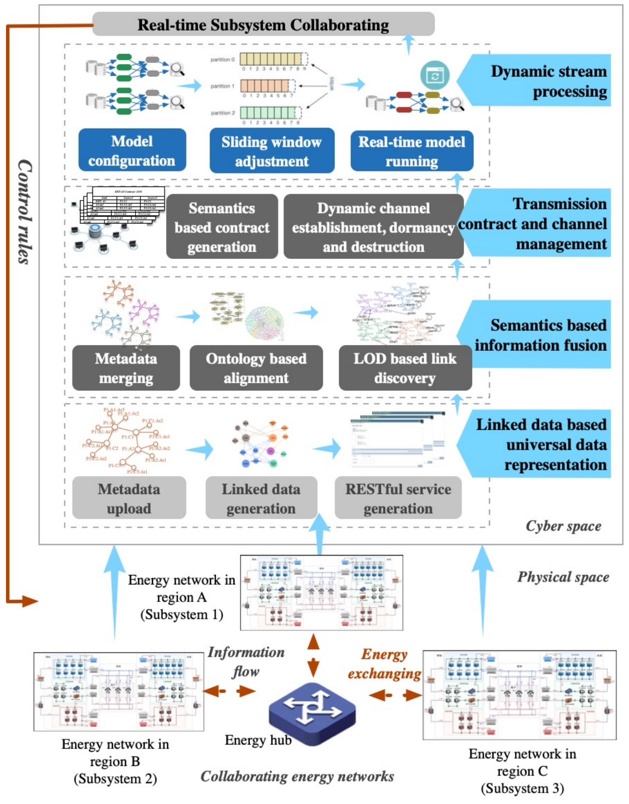

Real-time Subsystem Collaborating
Control rules
Dynamic stream processing
Model configuration
Sliding window adjustment
Real-time model running
Transmission contract and channel management
Semantics based contract generation
Dynamic channel establishment, dormancy and destruction
Semantics based information fusion
Metadata merging
Ontology based alignment
LOD based link discovery
Linked data based universal data representation
Metadata upload
Linked data generation
RESTful service generation
Cyber space
Physical space
Energy network in region A (Subsystem 1)
Information flow
Energy exchanging
Energy hub
Energy network in region B (Subsystem 2)
Energy network in region C (Subsystem 3)
Collaborating energy networks

Fig. 1. The technology framework of the proposed approach.

a) Phase I: Linked data based universal data representation: This phase designs a set of linked data concepts as the universal representation and transforms the metadata of subsystems to linked data accordingly. Standard RESTful APIs are generated for uniformly access through the web as a digital twin for the physical system.

b) Phase II: Semantics based information Fusion: In this phase, metadata in linked data are aligned with the assistance of domain ontology. New potential links and related external data are discovered with the help of Linked Open Data. A global metadata graph is established.

c) Phase III: Transmission contract and channel management: Information transmission contracts are generated according to discovered relations among metadata. And corresponding information exchanging channels are established. When changes occur in metadata or data transmission, the channels are on-demand established, dormant or destructed and the network dynamically evolves.

d) Phase IV: Dynamic stream processing: According to the application models defined in linked data, the corresponding input data are linked and listened in channels. A sliding window is adjusted for each model according to linked data. When the width is settled, the models are dynamically updated to run real-time processing.

IV. The Approach

An Integrated energy system is complicated containing various energy network subsystems. The information collaboration between them is difficult because they are black boxes to each

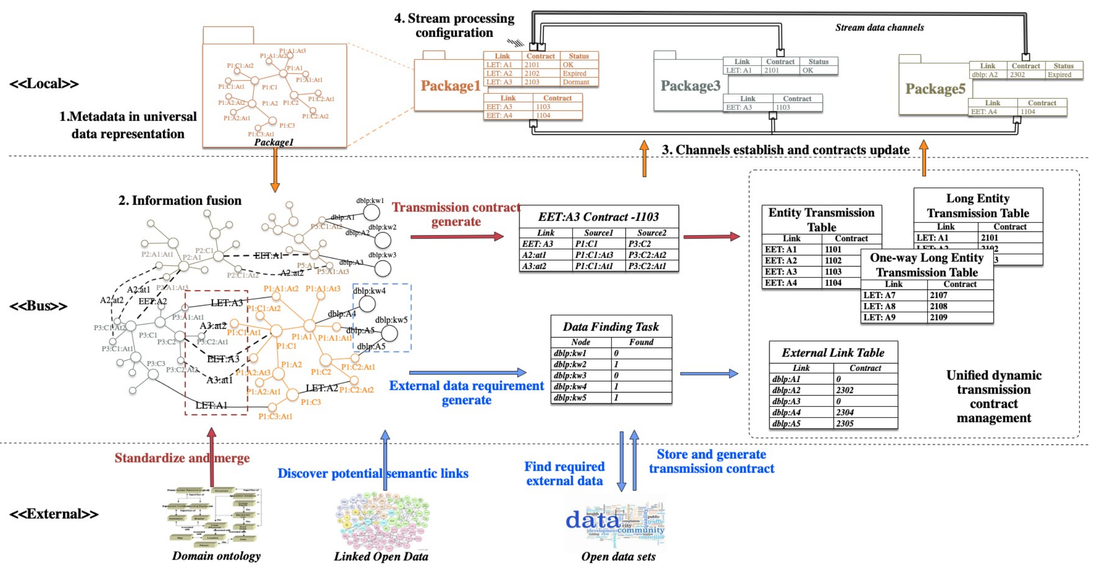

<<Local>>
1.Metadata in universal data representation
4. Stream processing configuration
Stream data channels
Package1
Package3
Package5
3. Channels establish and contracts update
<<Bus>>
2. Information fusion
Transmission contract generate
EET:A3 Contract -1103
Data Finding Task
External data requirement generate
Entity Transmission Table
Long Entity Transmission Table
One-way Long Entity Transmission Table
External Link Table
Unified dynamic transmission contract management
<<External>>
Standardize and merge
Discover potential semantic links
Find required external data
Store and generate transmission contract
Domain ontology
Linked Open Data
Open data sets

Fig. 2. The roadmap of the proposed approach. Starting with metadata uploading, a global metadata graph is built in the bus using semantic technicals. Information transmission contracts are generated according to semantic links and indexed in corresponding tables. On-demand information exchanging channels are managed according to the table. And stream processing is implemented at the end of channels.

other with separate data storage, control logic and functions. Fig. 2 illustrates the roadmap of the proposed approach to solve the issue. First, metadata in universal data representation is uploaded to a shared bus. Then, data merging is done to align multi-source data on the basis of domain ontologies and LOD. And new potential relations are discovered for transmission contract construction. Transmission channels are managed under the direction of contracts. Stream processing is implemented at the channels and adjustment of time windows is done with data monitoring.

A. *Linked data based universal data representation*

Considering the features of energy IoT data, a universal data representation is designed. First, concerning the heterogeneity in data schemas and structures, linked data is utilized to adopt an arbitrary number of attributes and relational links. A typical linked data tuple $\langle s, p, o\rangle$ model is applied, where s and o refer to the subject and object instances, and p refers to the predicate that records the relation from s to o. From the utility aspect, data can be divided into online ones that record the running status of a system and offline ones record the static information. Also, according to the source of data, internal data generated by the system and external data accessed from the Internet or other systems are also different. Therefore, a linked data based metadata model is designed as shown in Fig. 3, four basic and two utility concepts are defined for uniformly model metadata. And a part of instances in an EI system is shown as an example.

Device defines the equipment in IoT from the edges to the centres, including stations, terminals, sensors, etc.. Each *device* has three basic kinds of predicates: *hasInFlow*, *hasOutFlow* and *hasProperty*. The former two link to *Flows* the device leverages and generates respectively. *hasProperty* links the static attributes, such as size, position, working temperature, etc.. A *Property* can be a numerical, string,

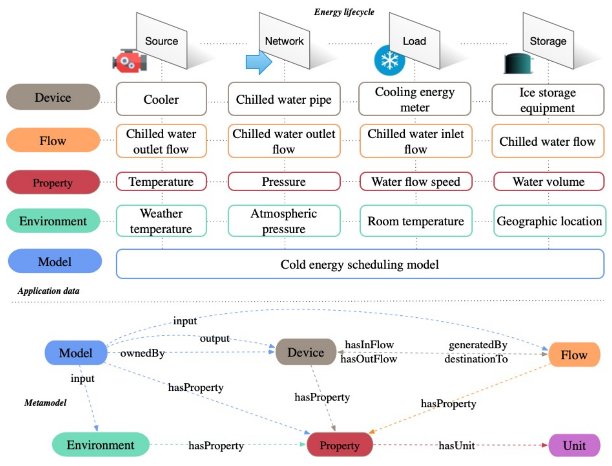


Fig. 3. Basic entities and relations in the metamodel as linked data with metadata instances of close relations in energy systems. The above part contains the 4 stages and the typical data correspondingly. The bottom part shows the predefined 6 concepts and interrelations.

or any complex structure. As shown in Fig. 3, a cooling machine `Cooler` owns tuple data $\langle Cooler, isA, Device\rangle$, $\langle Cooler, hasOutFlow, Chilled\_water\_outlet\_flow\rangle$, etc..

Flows are transaction data [18] record status of devices. They take *generatedBy* and *destinationTo* to refer the devices to generate and receive the flow data. *hasProperty* contains the attributes of a data flow. In Fig. 3, `Chilled water outlet flow` has typical tuples like $\langle Chilled\_water\_outlet\_flow, isA, Flow\rangle$ and $\langle Chilled\_water\_outlet\_flow, hasProperty, Temperature\rangle$ For each numeric property, a *unit* is applied. For example, $\langle Temperature, hasUnit, F\rangle$Units can help us quickly recognize the category of the data and align data from different sources.

Environment is a special kind of flow not recorded by the devices in the system but introduced from outsides such as the Internet. It has strong influences on the running of systems. For example, the daily temperature is not recorded in an energy system but is an influencing factor in the prediction of consumed thermal volume, which makes it an *environment* entity. It has no *generatedBy* and *destinationTo* predicates.

*Model*s are the logically defined analysis models that neither have a corresponding physical entity nor a flow generated by any physical entities. The models carry out services, analysis and prediction models to regulate the running of the physical system. The predicate *output* connects the generated decision data flows and destination devices. Similarly, *input* refers to the data used in the analysis model. *hasProperty* here is for the fixed parameters applied in the model, and *ownedBy* links to the subsystems where the model runs. In Fig. 3, `Cold energy scheduling model` is used to control the cold energy balance in the system, and involves all the flows as input and indirectly linked with the others in the whole energy lifecycle to get a more precious result.

According to the updating frequencies of data, we define models and devices as *offline* data for the data are fixed since design time and are rarely exchanged during runtime. While data flows and environments are *online* because they arise and grow along with system running.

Layered namespaces are applied to demonstrate the belonging of data and also compose a part of the resource's unique identifier. By applying the linked data metamodel, data instances can be stored in a graph database and accessed by RESTful APIs with unique resource identifies (URI).

B. Semantics based information fusion

To find the related data and construct flexible channels, we need to fuse the data according to the semantics. We first match the same entities by similarity calculation, including the textual and the structural parts, as shown in Algorithm.1. The textual similarity Sim_{lex} utilizes the full name of the entity by separating the layered namespace into fragments and applying Jaccard similarity on the fragment sets as in Line 3. However, the fragments of both entities are not strictly the same but also with lexical similarities. To that end, the union set is generated according to Line 2 in Algorithm.1 on the basis of the lexical similarity of fragments. The lexical one is the cosine similarity of word embeddings [19] $e(v)$ of two elements. As for the structural similarity, we calculate the Jaccard similarity of their object sets in Line 5, where $Obj(v)$ denotes the set of all vertices that is the object of subject v in a tuple among the metadata. Similarly, the objects themselves have similarities, so the textual similarities of objects are used to calculate union in Line 4. And the combined similarity sums up the textual and the structural similarity with two weights in Line 6. When the similarity exceeds a threshold, we consider the two data entities are the same.

Based on the vertices matching algorithm, the $sameAs$ relations are found. Then, we introduce the domain ontology to find potential relations among them. We match the entities in metadata with the ones in domain ontology using Algorithm.1 and when the combined similarity exceeds the threshold, we

Algorithm 1 Vertices matching algorithm

```
Input: two vertexes $v_1$, $v_2$
Output: Combined similarity $Sim_{com}$
 1: function CALCULATESIMILARITY($v_1, v_2$)
 2:     $union_{txt} \leftarrow Union(Frag(v_1), Frag(v_2), 'lex', \theta_{lex})$
 3:     $Sim_{txt} = \frac{|Union_{txt}|}{|Frag(v_1)|+|Frag(v_2)|-|Union_{txt}|}$
 4:     $union_{strct} \leftarrow Union(Obj(v_1), Obj(v_2), 'txt', \theta_{txt})$
 5:     $Sim_{strct} = \frac{|Union_{strct}|}{|Obj(v_1)|+|Obj(v_2)|-|Union_{strct}|}$
 6:     $Sim_{com} = \alpha Sim_{txt} + \beta Sime_{strct}$
 7:     return $Sim_{com}$
 8: end function
 9: function UNION($Set_1, Set_2, type, \theta$)
10:     for $n \in Set_1$ do
11:         for $m \in Set_2$ do
12:             $Simlist.append(n, m, Sim_{\$type\$}(n, m))$
13:         end for
14:     end for
15:     Descending order $Simlist$
16:     while $Simlist$ not null do
17:         if $Simlist[0] > \theta$ then
18:             $Union(Set_1, Set_2).apend(Simlist[0])$
19:             Remove elements with $n$ and $m$ from $Simlist$
20:         end if
21:     end while
22:     return $Union(Set_1, Set_2)$
23: end function
```

add the $standardName$ predicate for the vertex to align the concepts. Further, the direct relations and the entity in the domain ontology are added to metadata and recorded as the initial accordance of data transmission contracts.

When the found link connects two instances of *Property*, it is meaningless towards transmission because the transmission of the single attribute lacks useful identifier information such as a primary key or a timestamp. To solve the problem, we upcast the link to add necessary assistant information to make the transmission meaningful. To do so, we first decide whether the link is useful by the concept of IEF(Inverse Entity Frequency). It shows the frequency of the term as the attribute of entities in the whole metadata base.

$$ief_t = \log \frac{|E|}{|\{s : (?s, ?p, t) \in Edge\}|} \quad (1)$$

where $|E|$ denotes the total number of entities that do not have a corresponding edge vertex in the metadata base, s denotes the entities that own the attribute t. The ief measures the importance of a property, when ief exceeds a threshold θ, we believe that the attribute has significance and the link is moved to the subject of the attribute, which is called **upcast**. Otherwise, the link is ignored, which is called **rollback**. A motivation example is $timestamp$ which owns by most of the data but is meaningless to transmit alone. By using IEF, these kinds of useless links can be omitted.

Besides the direct relations discovered from data merging, there are also potential relations with external elements that have an influence on existing data. To further enrich the semantic information and find more potential relations, LOD is utilized. Similar to discovering links from the domain ontology, the $sameAs$ relation is first built linking to LOD entities. Then, a step length is set to find whether there is another coordinate reachable within the step length. If so, a relation $potentialRelatedTo$ will be built between the corresponding entities. For the directly connected entities in LOD but not in the local metadata, it will be added in the metadata. And a data finding task is registered and broadcast to administrators to find open data of that entity from the Internet. Once the data are found and accessible, an $Environment$ entity is added.

C. Transmission contract and channel management

As the data scale grows large in IoT systems, the automated data exchanging is important. To dynamically configure the data exchanging channels, the transmission contract is proposed. It contains the metadata matching result between two components of different subsystems, aiming to notify the two components which data they can access and how to transform and utilize from the other side. A transmission contract is denoted as

$$\begin{aligned} contract = &\langle name, type, commonAttr, \\ & relatedAttr, sAttr, oAttr\rangle \end{aligned} \tag{2}$$

where

$$\begin{aligned} & name := (p, s, o), \\ & commonAttr := \{(name, s.attr, o.attr)\}, \\ & relatedAttr := \{(p, s.attr, o.attr)\}, \\ & sAttr := \{s.attr\} \backslash (commonAttr \cup relatedAttr), \\ & oAttr := \{o.attr\} \backslash (commonAttr \cup relatedAttr) \end{aligned} \tag{3}$$

in which $type$ denotes the category of the contract, $commonAttr$ denotes the set of same attribute pairs between two entities, $relatedAttr$ is for other linked attributes, $sAttr$ is for attributes only in the entity s and so as $oAttr$. The p in $name$ denotes the predicate of the link, s and o are the subject and object entities. The $name$ in $commonAttr$ is the standard entity name in domain ontology. The attributes of a entity are the direct object nodes linked to the specific entity.

According to the data types of the subject and the object of the link, the transmission contracts are divided into three categories as illustrated in Table I. The entity transmission contract is for offline data entities. Since the updating and transmitting of offline data are pulsed, the connection is built on demand without keeping a dedicated channel. On the contrary, online data are generated consequently and require high efficiency and stability in reading and writing. Therefore, for online data entities, the long transmission contracts are used to build a persistent connection. Analogously, the one-way long transmission contract is used for the connection between online and offline data. The data flows from online data to the offline data are persistent and the reverse ones are immediate. Three indexing tables are maintained in the bus to store the three kinds of transmission contract entries.

Exchanging channel establishment. A transmission contract is sent to both sides with the IP address of the other side. Two tables are maintained at each subsystem to direct

TABLE I
THREE TYPES OF TRANSMISSION CONTRACTS

Type	Subject	Object	Connection
Entity transmission contract	Offline	Offline	Immediate
Long transmission contract	Online	Online	Persistent
One-way long transmission contract	Online	Offline	Persistent(Forward)
	Offline	Online	Persistent(Reverse)

the establish and run of the channels. One is for persistent connections, and the other is for immediate connections. For the persistent connection, a private channel, ie. an independent process with a dedicated queue is built on both sides for data transmission listening and handling. In the pull mode, the requester refers to the contract to locate the data. The listener in the sender side checks the IP address according to the transmission contract and send data. On the requester side, the listener utilizes a dedicated queue to receive the data. In the push mode, the sender checks the contract table to find the destinations of the data and aligns data accordingly before sending. The receiver listener checks the data and IP and stores the data in the specific queue. A pair of processes that handle the same contract compose a channel. Each persistent connection utilizes a private channel. The fixed bandwidth is assigned to the channel to make sure the steady transmission of the persistent connection. For the immediate connections, the listener process and queue are shared. In such a channel, data from different sources compete for the resources and the bandwidth, which is called a public channel.

Exchanging channel destruction. When the metadata change, transmission contracts can be expired. The corresponding system nodes will be informed, and the contract tables will be updated. If a (one-way) long entity transmission contract is expired, the established data exchanging channel will be released to save the computing resources for others.

Exchanging channel dormancy. In some cases, the metadata does not change, but the data channels are still rarely used because (1) the two nodes connected has not yet realized they need that data or (2) the data are intermittently used. Under these circumstances, a channel dormancy mechanism is applied. When the listener finds no data request for a lasting time while the common channel has a pending request, the private channel will be dormant to release the computing resources and lower the fixed bandwidth for the other running channels. And when data request comes again in the private channel, the channel will be awakened. In this way, we keep a priority of the data transmitted in the private channel and also a better performance of the whole system.

By applying the above data transmission strategy, the metadata modification of a single node can be dynamically accepted by all related nodes, namely, the extensibility is guaranteed. Moreover, the unrelated nodes are still black boxes to others, which keeps the reliability and safety of the system.

D. Dynamic stream processing

Analysis models are bound to subsystems by system users. Fig. 4 shows the process of running real-time analysis with stream processing. A sliding window receives the input data defined by the model from corresponding exchanging channels. Then, data preprocessing is performed in the sliding windows of time t, and the model utilizes the data in t to calculate and analyze the real-time result. Obtaining the results, the models generate real-time output flows. The new output flows are then sent to the destination subsystems via the specific private channels according to contracts. Finally, the output flows are received by destination subsystems and new models are real-timely performed.

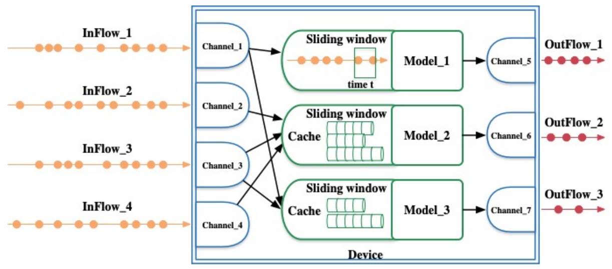


Fig. 4. The mechanism of real-time analysis with stream processing.

When applying the models, to improve the efficiency and performance of the models, we divide them into two kinds: timing-related model and timing-unrelated model. For the timing-related models, the data transmission via the network can lead to data disorder. Therefore, we reorder the received data in the sliding windows. A cache is applied for each time-related flow in the sliding window. While for the timing-unrated model, a single time window is enough for data preprocessing. Besides the reordering of the data, there are other data preprocessing to be done in the sliding windows, such us standardizing of units. By introducing open data, the conversion relations among units are added to the linked data, and the automatic data unit alignment is performed in the sliding window. Considering the width of time windows is decided by the usage of the models but the manual setting for every model is costly for a large amount of the models in a system. An automated window width adjustment mechanism is provided by derivative prediction. When the result has an acute change, the width of the window is shortened. By setting the time window, the models are applied to small bundles of data and results is attained in a short time. According to the transmission contracts, the results are accurately transmitted with high priority to fulfill the real-time control.

V. Case Study and Discussion

In this section, we use an integrated energy system with 3 regional smart energy networks as a case to show the application of the framework and discuss the approach with the state of the arts to illustrate the characteristics of the proposed framework.

A. Case Study in City-wide Integrated Energy System

1) Case Description: The integrated energy system is composed of 3 regional smart energy networks and an energy hub for energy exchange. The 3 energy networks located in different districts in city S. These regional subsystems were deployed on private clouds as independent EI systems. Each manages the whole energy lifecycle of a region, ie. energy supply source, transmission network, load and storage. The energy is multidimensional including cold energy, thermal energy and electricity. To achieve the city-wide energy optimal utilization, the energy hub joins to transform 3 types of energy among the 3 subsystems. From the information aspect, there are hundreds of devices and data flows in each subsystem as shown in Table. II. And there are some analysis models deployed in each subsystem for different application purpose. For the detailed design of analysis models are not the focus of this paper, a simplified application model is used to demonstrate the usage of the framework. An implementation of the framework of the case system is shown in Fig. 5.

TABLE II
THE RESOURCE AMOUNT OF THE INTEGRATED ENERGY SYSTEM

	Flows	Devices	Models	Total
Subsystem 1	398	397	37	832
Subsystem 2	274	163	24	461
Subsystem 3	288	176	23	487
Total	960	736	84	1780

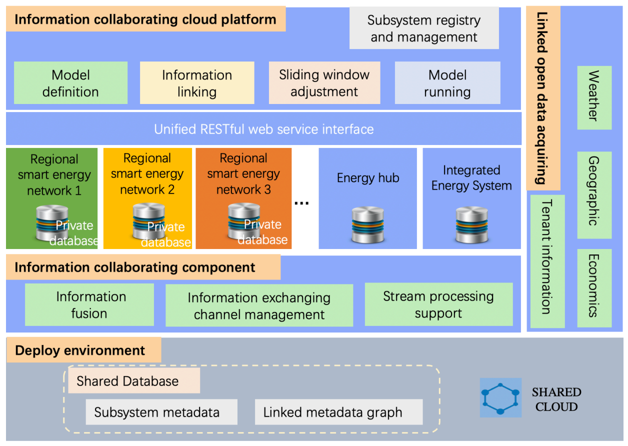

Information collaborating cloud platform
Subsystem registry and management
Model definition
Information linking
Sliding window adjustment
Model running
Unified RESTful web service interface
Regional smart energy network 1
Private database
Regional smart energy network 2
Private database
Regional smart energy network 3
Private database
...
Energy hub
Integrated Energy System
Information collaborating component
Information fusion
Information exchanging channel management
Stream processing support
Deploy environment
Shared Database
Subsystem metadata
Linked metadata graph
SHARED CLOUD
Linked open data acquiring
Tenant information
Weather
Geographic
Economics

Fig. 5. An implementation of the information collaborating platform based on the proposed framework.

2) Application Scenarios: To better demonstrate the application of the methodology, we illustrate scenarios of two kinds of energy collaborating as shown in Fig. 6. The cloud platform is the entry of open data from the Internet including weather, locations and economics. For normal energy collaborating, it sends the open data to 3 subsystems according to the *external link table*. An energy load prediction model runs in each subsystem, leveraging the open data with internal data to predicate its energy load and supply ability in next hour. The predicted results are sent to the cloud platform according to the entity transmission tables to run the global scheduling model. The model gets data from the 3 subsystems in a sliding window, and analyze if energy exchanging is needed. If so, energy transmission rules are sent to the hub for execution. The other example is an emergency-triggered energy exchanging process. When some energy source devices encounter errors inducing the energy supply interruption in some area, the subsystem sends the error information to the energy hub informing the lack in energy supply. The platform receives and runs the error responding model to request left supply abilities and distances from other subsystems. A new schedule is made to transform energy from the normal subsystems to the affected area.

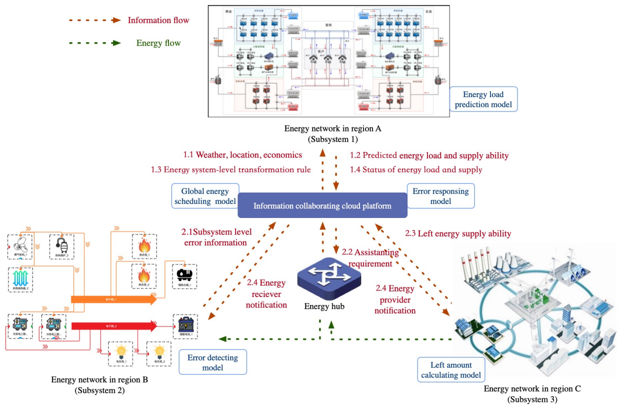
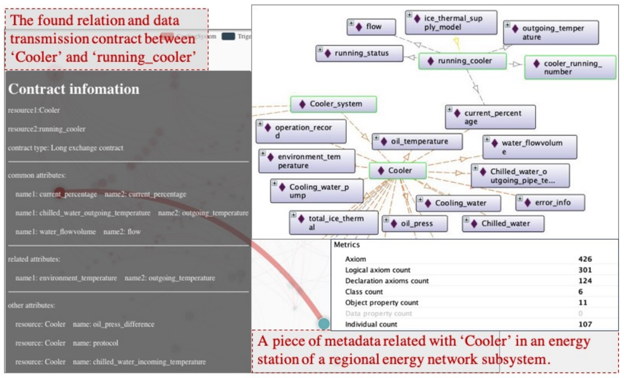

Information flow
Energy flow
Energy load prediction model
Energy network in region A (Subsystem 1)
1.1 Weather, location, economics
1.2 Predicted energy load and supply ability
1.3 Energy system-level transformation rule
1.4 Status of energy load and supply
Global energy scheduling model
Information collaborating cloud platform
Error responsing model
2.1Subsystem level error information
2.2 Assistant requirement
2.3 Left energy supply ability
2.4 Energy reciever notification
Energy hub
2.4 Energy provider notification
Error detecting model
Left amount calculating model
Energy network in region B (Subsystem 2)
Energy network in region C (Subsystem 3)

Fig. 6. The system composition and example information flow of the integrated energy system based on the proposed framework.

To achieve the above processes, the subsystems are firstly registered on the platform. Fig. 8 shows a part of registered metadata of a regional energy network subsystem. A semantic network is incrementally built by information fusion. And the transmission contract is generated.



Contract infomation

resource1:Cooler

resource2:running_cooler

contract type: Long exchange contract

common attributes:

name1: current_percentage name2: current_percentage

name1: chilled_water_outgoing_temperature name2: outgoing_temperature

name1: water_flowvolume name2: flow

related attributes:

name1: environment_temperature name2: outgoing_temperature

other attributes:

resource: Cooler name: oil_press_difference

resource: Cooler name: protocol

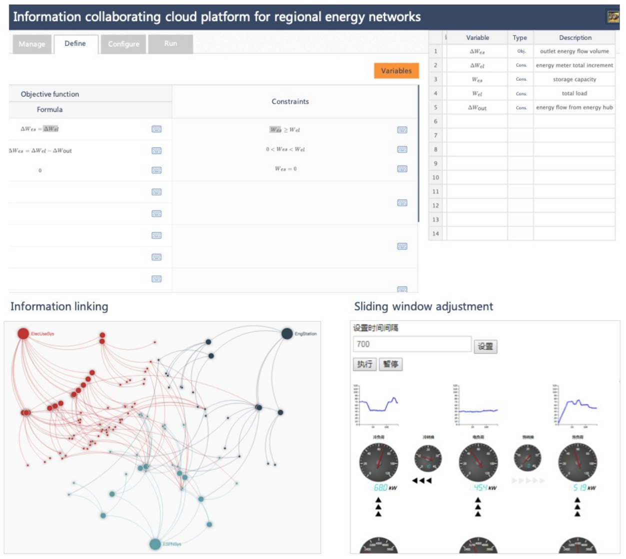
resource: Cooler name: chilled_water_incoming_temperature

Metrics	
Axiom	426
Logical axiom count	301
Declaration axioms count	124
Class count	6
Object property count	11
Data property count	0
Individual count	107

Fig. 7. A piece of metadata in universal data representation of a 'Cooler' in an energy station in a subsystem and an instance for exchanging contract.

Secondly, a model is defined on the platform by formulating the model as codes. A simplified energy scheduling model with only internal energy load and storage can be expressed as in Fig. 8 where the web pages of the cloud platform are shown including define models, fuse information, adjust sliding windows and finally run models.

To run the model, an adjustment on the sliding window is done. Leveraging the historical data running on the model, we get the results in Fig.9 showing the data trend predicting error rates in different predicting methods. The method 1-6 sum derivative and the last data. The method 7-9 calculate the mean of the data in a window as the predicted data. The



Information collaborating cloud platform for regional energy networks

Objective function: Formula	Constraints
$\Delta W_{es} = \Delta W_{el}$	$W_{es} \geq W_{el}$
$\Delta W_{es} = \Delta W_{el} - \Delta W_{out}$	$0 < W_{es} < W_{el}$
0	$W_{es} = 0$

	Variable	Type	Description
1	ΔW_{es}	Obj.	outlet energy flow volume
2	ΔW_{el}	Cons.	energy meter total increment
3	W_{es}	Cons.	storage capacity
4	W_{el}	Cons.	total load
5	ΔW_{out}	Cons.	energy flow from energy hub
6			
7			
8			
9			
10			
11			
12			
13			
14			

Fig. 8. The screen shots of cloud platform for information collaborating.

method 10-15 sum the difference and the mean. Except for the methods with descending results, the best results are at window width 2 and 3. In this way, the proper window width of 2 is applied in this model. After all these configurations, the model can run to give control to the whole system.

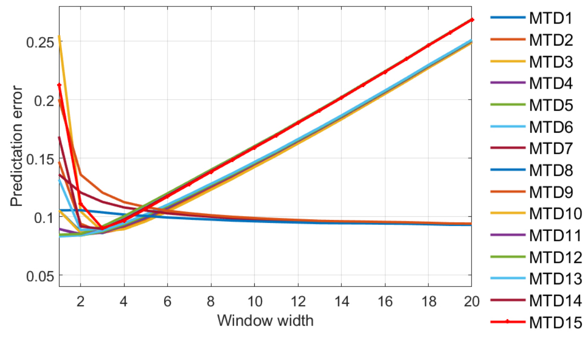


Fig. 9. The window width choice for stream processing in the case. The window width of 2-3 is the best choice with the lowest average error.

B. Discussion

We compare our approach with three existing data communication and processing solutions in IoT systems from the aspect of methodology and performance as shown in Table III.

LEDM [20] refers to the Linked Enterprise Data Model and its use in IoT system. The data model is derived from linked data, and data alignment is conducted according to domain ontology for heterogeneous data management. A stream-based ETL process is applied in data management. IoT-ECS [21] is a real-time data analytics and event detection for IoT based communication system. It focuses on efficient and dynamic ad-hoc stream data communication, leveraging existing ontologies for data representation and stream processing for data fusion and semantic analysis. SEMIoTICS [15] is a supervisor to build semantic models for IoT components. It utilizes SSN ontology as the semantic annotations for IoT data and applies semantic reasoning to configure the components according to feedback data.

TABLE III
COMPARISON WITH RELATED WORK

Feature	LEDM [20]	IoT-ECS [21]	SEMIoTICS [15]	Our framework
Data representation	Linked Enterprise Data Model	Combination of existing ontologies	Combination of existing ontologies	Universal Data Representation and REST
Information fusion	Domain ontology aided	Domain ontology aided	Reasoning based	Domain ontology and LOD aided
Data transmission	Not mentioned	Dynamic direction selection	Semantic directed, automatic and dynamic	Semantics directed, automatic and dynamic
Stream Processing	Supported	For linked data reasoning and query	Not mentioned	Supported
Adaptability	Domain-limited data fusion	Domain-limited data representation	Domain-limited data representation	LOD aided data fusion
Feasibility	Limited	Weak	Fair	Good
Flexibility	Weak, fixed	Fair, dynamic runtime update	Good, automatic runtime update	Good, automatic runtime update

From the perspective of functionalities, linked data and ontologies are the main choices for **data representation**. In the **information fusion** part, LEDM and IoT-ECS take domain ontologies as reference and SEMIoTICS is based on reasoning. Our framework introduces linked open data except for domain ontologies making the fusion less domain specific. In **Data transmission** part, LEDM does not consider the communication issue. IoT-ECS dynamically manage transmission via direction selection at every moment data generated. SEMIoTICS and our framework leverage automatically generated semantic network to direct and dynamically update on changes. For the **stream processing** aspect, except for SEMIoTICS, the other three all implement stream processing to support the real-time analysis. However, IoT-ECS only support stream semantic reasoning and query. And SEMIoTICS only applies stream processing in configuration but not in run time. LEDM and our framework implement real-time control of the whole system based on stream processing.

As for the performance measures, the **adaptability** is the universalness and expandability of the system. IoT-ECS and SEMIoTICS depend on existing domain ontologies for data representation decreasing the versatility. LEDM and our approach use domain-specific data fusion approaches, which also harms versatility. However, LOD utilization weakens domain dependence.

The **feasibility** measures how systems work in a practical environment. The lack of real-time control and the dependency on domain ontology weaken the feasibility of IoT-ECS. LEDM ignores data communication in the system, which makes it hard to be applied to complex systems. SEMIoTICS uses semantic data fusion to direct the data transmission and system configuration, but the lack of stream processing in system run and the domain ontology-based data representation limit the scalability. While our approach comprehensively considers the information in full physical lifecycle and real-time processing, which is most feasible among the four solutions.

The **flexibility** is the ability to face internal and external changes [22]. LEDM is not acquainted with any new connections and thus hard to extend. IoT-ECS dynamically adapts the changes, however, it cannot inform data receivers about the changes in real-time, which is weaker than SEMIoTICS and our approach where data transmission is dynamically changed on the basis of data semantics.

VI. CONCLUSION

In this paper, a stream processing framework based on linked data is proposed for information collaborating of multiple regional energy networks. The framework first defines a universal data representation based on linked data, which provides a unified way to describe heterogeneous data in IoT with semantic information. Then domain ontologies and Linked Open Data are introduced to build a rich semantic model of the metadata to link related data. By applying semantics-based data fusion, information transmission contracts are discovered automatically and dynamically. Accordingly, flexible information exchanging channels are established for different types of data. Stream processing is applied for online data analysis, making information collaborating become real-time resulting, efficient and understandable. The adaptability, flexibility and feasibility are demonstrated by the case study in a city-level multi-dimensional energy integration system.

In the future, we will take fog computing into the framework to build a layer model for data fusion to improve the performance of transmission contract discovery. Also, we will work on the approach of automatically finding external data to make the framework better utilize the open data to provide stronger Internet insights for processing.

REFERENCES

[1] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, and J. Wu, "A survey on energy internet: Architecture, approach, and emerging technologies," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2403–2416, Sep. 2018.

[2] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[3] M. Dias de Assuno, A. da Silva Veith, and R. Buyya, "Distributed data stream processing and edge computing: A survey on resource elasticity and future directions," *Journal of Network and Computer Applications*, vol. 103, p. 117, Feb 2018.

[4] D. Huo, C. Gu, K. Ma, W. Wei, Y. Xiang, and S. Le Blond, "Chance-constrained optimization for multienergy hub systems in a smart city," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1402–1412, Feb 2019.

[5] C. Bizer, T. Heath, T. Berners-Lee *et al.*, "Linked data-the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.

[6] P. Carbone, S. Ewen, G. Fóra, S. Haridi, S. Richter, and K. Tzoumas, "State management in apache flink&reg;: Consistent stateful distributed stream processing," *Proc. VLDB Endow.*, vol. 10, no. 12, pp. 1718–1729, Aug. 2017.

[7] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, and D. Ryaboy, "Storm@twitter," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 147–156.

[8] R. Dautov, S. Distefano, D. Bruneo, F. Longo, G. Merlino, and A. Puliafito, "Data processing in cyber-physical-social systems through edge computing," *IEEE Access*, vol. 6, pp. 29 822–29 835, 2018.

[9] D. Puschmann, P. Barnaghi, and R. Tafazolli, "Adaptive clustering for dynamic iot data streams," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 64–74, Feb 2017.

[10] M. Compton, P. Barnaghi, L. Bermudez, R. Garca-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, and A. Herzog, "The ssn ontology of the w3c semantic sensor network incubator group," *Web Semantics Science Services Agents on the World Wide Web*, vol. 17, no. 4, pp. 25–32, 2012.

[11] D. Le-Phuoc, H. N. M. Quoc, H. N. Quoc, T. T. Nhat, and M. Hauswirth, "The graph of things: A step towards the live knowledge graph of connected things," *Journal of Web Semantics*, vol. 37, pp. 25–35, 2016.

[12] S. Chun, S. Seo, B. Oh, and K.-H. Lee, "Semantic description, discovery and integration for the internet of things," in *Semantic Computing (ICSC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 272–275.

[13] M. Wylot, P. Cudr-Mauroux, M. Hauswirth, and P. Groth, "Storing, tracking, and querying provenance in linked data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1751–1764, Aug 2017.

[14] M. Ganzha, M. Paprzycki, W. Pawowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the internet of things; an overview from the inter-iot perspective," *Journal of Network & Computer Applications*, vol. 81, 2016.

[15] G. M. Milis, C. G. Panayiotou, and M. M. Polycarpou, "Semiotics: Semantically enhanced iot-enabled intelligent control systems," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1257–1266, Feb 2019.

[16] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[17] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, Sept 2016.

[18] D. Riboni, L. Pareschi, and C. Bettini, "Js-reduce: Defending your data from sequential background knowledge attacks," *IEEE transactions on dependable and secure computing*, vol. 9, no. 3, pp. 387–400, 2012.

[19] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 1555–1565.

[20] K. Taneja, Q. Zhu, D. Duggan, and T. Tung, "Linked enterprise data model and its use in real time analytics and context-driven data discovery," in *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 2015, pp. 277–283.

[21] M. I. Ali, N. Ono, M. Kaysar, Z. U. Shamszaman, T.-L. Pham, F. Gao, K. Griffin, and A. Mileo, "Real-time data analytics and event detection for iot-enabled communication systems," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 42, pp. 19–37, 2017.

[22] E. Lannoye, D. Flynn, and M. O'Malley, "Transmission, variable generation, and power system flexibility," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 57–66, 2015.