Prediction of Telecommunication Network State Based on Knowledge Graphs

Kirill Krinkin, Igor Kulikov, Alexander Vodyaho
Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
kirill@krinkin.com, i.a.kulikov@gmail.com,
aivodyaho@mail.ru

Nataly Zhukova
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St. Petersburg, Russia
nazhukova@mail.ru

***Abstract*— The article focuses on developing a new prediction method for telecommunications network state based on knowledge graphs. The article analyses the prediction methods supported by current network monitoring and management systems. The proposed prediction method is based on analysis of behavior of each end-user device that allows reach high accuracy of prediction. Knowledge graphs of the networks are build using domain knowledge and statistical data. They contain knowledge about the possible states of the elements of the networks along with the knowledge about possible transitions between them that can be presented in the form of prediction rules. These rules allow predict the state of the network for the defined timestamp in the future. In the case study the practical task of internet traffic prediction for a segment of cable TV operator network is considered. Conclusions are formulated, and the areas of further research are defined.**

I. INTRODUCTION

A. *The task of prediction for telecommunication networks*

Task of prediction states of the networks in the telecommunication domain is an actual task for telecommunication operators. The most actual parameter to be predicted from operators' point of view is the network load. Information about the expected level of the network load can be used for network reconfiguration that allows avoid situations like "out of services" for network subscribers. In modern telecommunication networks (TN), the main volume of traffic is the internet traffic from subscriber devices. Traffic depends on user behavior, types of communication services and network configurations. In addition, it is different for different time of day, work / weekend, seasons. Also, there is a dependency on various sports, cultural, political and other events, usually they cause the increase of traffic load. The paper proposes a solution for predicting state of a telecommunications networks using knowledge graphs.

B. *Traffic prediction methods overview*

The following models can be used for traffic prediction [1]:

- **Linear Time Series model.** Linear Time Series models allow describe the covariance of the time series. There are two popular sub-groups of linear time series models: Auto Regressive (AR) and Moving Average (MA) models, which are combined in Auto Regressive Moving Average model. Linear time series models are traditionally used for network traffic prediction [2], [3], [4].
- **Nonlinear time series model.** Nonlinear time series are generated by non-linear dynamic processes. They exhibit features that can't be modeled by linear functions such as time-change variance, asymmetric cycle, higher-moment structures, thresholds and breaks. For nonlinear prediction of networks states such techniques as neural networks, fuzzy logic are used [5], [6], [7].
- **Hybrid model.** The hybrid model is a combination of two or more models. The hybrid model can provide higher prediction accuracy in relation to single models. Typically, the hybrid model is built as a combination of linear and nonlinear models. It can give good results in prediction of the network traffic level [8], [9], [10].
- **Decomposition model.** The time series are generally decomposed into four components. Each component is defined below [11].
    - Trend component - trend is long - term propensity, increase and decrease in the time series data. Trend component represents the structural variations of low frequency in time series.
    - Cyclical component - cyclical component indicates the medium-term fluctuation. The component displays periodic increase and falls.
    - Seasonal component - is variations in time series data that is influenced by seasonal factors such as year, quarter, month, week, day, hour. The seasonal component has stable variation intra time series data.
    - Irregular component - is a residual time series that is obtained after removing trend and seasonal components.

All the discussed methods based on time series use only historical statistical data. This limitation does not allow using this approach for traffic prediction in dynamic TN which structure, state and behavior are changing in time. In the article a more flexible method of internet traffic prediction based on network statistical data, available prior information about traffic changing and traffic sensitive network events is proposed.

II. Method of model synthesis

A. *Structure of the knowledge graph*

To provide ability of flexible traffic analysis a complex TN model in the form of knowledge graph is considered. The model contains static models of network topology that include network and users devices, network services and applications as well as necessary of billing and rights control components. All the static data is provided by existing TN systems. Also the TN model contains operative data from TN devices which is retrieved from monitoring systems or directly from network devices. A prior information about traffic changing (services and network events schedules) is provided by special TN systems and management systems.

There are hierarchical and horizontal links between the elements of the model that define the multilevel structure of the model. Hierarchical links are used to reflect network topology, hierarchies of regions and other logical hierarchies. Horizontal links allow define cross-model links. The proposed structure of knowledge graph for traffic prediction task solving is given in Fig. 1.

- **Hierarchy of network services.** The hierarhy contains the services and applications which can be run on user devices. Catalogue of services and applications has hierarchical structure (including set of categories and sub-categories).

  Catalogue of services and applications is represented in KG as set of RDF triples where categories or subcategories on upper level as RDF subjects are connected with subcategories, services and applications on lower level as RDF objects by RDF predicates – ":includes". KG nodes which are used to describe services and applications do not include information about TN elements parameters (including internet traffic).

To reflect the changes in the network state in time the following dynamic elements are created and added to the graph during prediction algorithm execution:

- **Prediction nodes.** Prediction nodes allow link any two KG nodes as RDF subject and object using various predicates. This ensures solving of prediction tasks for various scenarios. Information about predicted parameters and values as well as prediction timestamp is included to prediction nodes structure. Each prediction node structure has a unique identifier.

  For linking static KG nodes the RDF "Statements about Statements" is used. In addition to making statements about selected resources, RDF can be used for making statements about other RDF statements. It is possible to refer to these statements as higher-order statements. In order to make a statement about another statement, a model of the original statement should be already built. This model is a new resource to which additional properties can be attached [12]. This technique allows connect a user device as RDF "subject" and a used service as RDF "object" using the predicate type that is suitable for particular situation. For every parameter change at every prediction step, a new prediction node is added. Detailed description of the suggested construction is below:

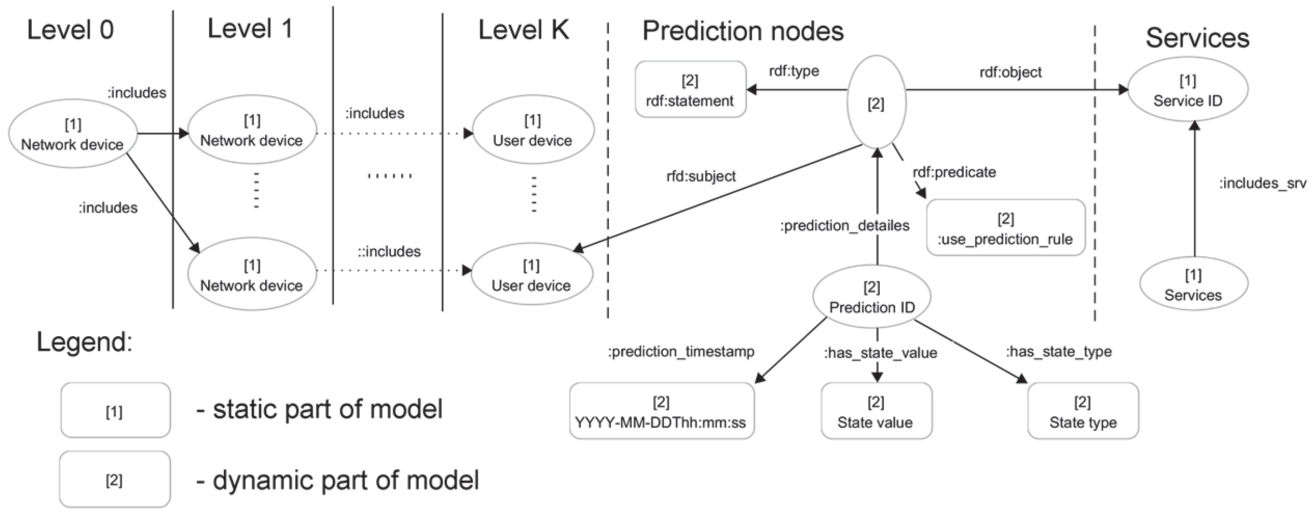

Fig. 1 Knowledge graph structure

The following static nodes define the KG structure:

- **Network devices hierarchy.** This hierarchy includes all the network switchers, routers, and end-user devices. The hierarchical links between them define the topology of the network. In the suggested structure all the network switchers, routers and other devices are assigned to the levels $0 \dots K-1$ hierarchy and all the user devices are assigned to the level K. From perspective of traffic prediction, traffic for every network element on levels $0 \dots K-1$ can be calculated as sum of traffic produced by the user devices connected to them.

  The network hierarchy is represented in KG as set of RDF triples where devices on upper level as RDF subjects are connected with devices on lower levels as RDF objects by RDF predicate – ":includes". KG nodes which describe the network topology don't include information about TN elements parameters (including internet traffic).

- Prediction node– reflects the particular prediction step for the particular network element. Prediction node is identified by its ID (Prediction ID). It is linked to attributes: timestamp, parameter type and parameter value as RDF "object". Prediction nodes are linked to virtual nodes.
- Virtual node has type "Statement about Statement" and is linked to User device ID as RDF "subject" and to Service ID as RDF "object".

B. *Network state prediction task*

The state of a network, a set of its elements or separate elements can be described by sets of parameters, i.e. incoming internet traffic. Changes of the network state in time are described as time series of parameters values, in particular, time series of incoming / outgoing internet traffic. Let $\{P_{t_{0-2}}, P_{t_{0-1}}, P_{t_0}, \dots, P_{t_w}, \dots\}$ be the number of packets observed over adjacent time intervals, each interval length is t_d. The time series of differences of the traffic volume $\{W_t = P_t - P_{t-d}\}, (d \geq 0)$ are stationary processes with short memory [13].

From time series perspective, the task of network model prediction can be defined as following. Let's define timestamp t_0 as the current time or the time in the past at which the state of the NT is known and the timestamp t_w as the time in future at which the state of the network should be estimated.

The model of the network for timestamp t_0 is defined as $D_{t_0} = \{D_{t_0,1}, D_{t_0,2}, \dots, D_{t_0,N}\}$, where $D_{t_0,i}$ - the model of the element i for timestamp t_0. Prediction network state task can be defined as the task of finding the sequence of models $\{D_{t_0}, D_{t_0+t_d}, D_{t_0+2\times t_d} \dots, D_{t_w}\}$, where D_{t_i} – is the network model for the timestamp t_i, and D_{t_w} – is the network model for the timestamp t_w, t_i, t_w – the timestamps in the future, t_d – the prediction step.

C. *Model for prediction tasks solving*

The input data for network state prediction is the following:

- Model of the network that is built using static data about the TN networks and enriched with statistical and operational data.
- A subset of nodes which states are interesting to the user and should be predicted.
- Timestamp in future defined by the user for which it is necessary to predict the state of the network (t_w),
- Time step of the prediction (t_d).

The prediction task is the task of transforming model that reflects the network state for the current moment of time or a moment of time from the past (t_0) to the model of the network at the defined timestamp in future (t_w). The process of transformation can contain one or a sequence of steps. The number of steps is defined by the time interval from t_0 to t_w and the step of prediction t_d. Assume the *m*-th step is reached. At this step model D_m for the timestamp $t_0 + m \times t_d$ is build.

For transforming models transformation rules are used. Transformation rules allow moving a node or a group of nodes from one state to another [14], rules are defined using network statistic for network elements.

A rule R_j defines how the set of parameters that describes the state of the *i*-th network element $D_{m,i}$ can change from step m to step $m + 1$:

$$D_{m,i} \xrightarrow{R_j} D_{m+1,i}$$

Every prediction rule R_j consists of the following components:

- Condition of use – it can be time, logical or other type of conditions.
- Probability of changing the value of the network element parameter.
- The value of the parameter at $m + 1$ step.

Every prediction step, a rule can be used or not according to current conditions.

Example of a rule definition is below:

(Parameter_1)&&(Condition_1)&& ... &&(Condition_N) → *Parameter value (P=probability)*
...
(Parameter_2)&&(Condition_1)&& ... &&(Condition_N) → *Parameter value (P=probability)*
...

Let N is the number of model elements and assume we have built a model $D_{t_m} = \{D_{t_m,1}, D_{t_m,2}, \dots, D_{t_m,N}\}$ for the timestamp t_m. To transform the model to the model D_{m+1}, rules are applied model elements:

$$D_{t_m,1} \xrightarrow{R_1} D_{t_m+t_d,1}$$
$$D_{t_m,2} \xrightarrow{R_2} D_{t_m+t_d,2}$$
$$\vdots$$
$$D_{t_m,N} \xrightarrow{R_N} D_{t_m+t_d,N}$$

In case of one-level model, at each step the information about the changes of the parameters values is added to the TN model and no additional calculations are required. For multi-level model we have a bit different situation:

- Let the model contains $K + 1$ levels $(0 \dots K)$.
- Let the rules are defined only for the parameters values of the elements on level K.
- Model elements on levels $(0 \dots K - 1)$ are distributed between the levels in the following way: $(q_1^0, q_2^1, \dots, q_M^{K-1})$, where q – number of elements on the level $(0 \dots K - 1)$. Model elements on levels $(0 \dots K - 1)$ are hierarchically connected to elements on the level K.
- On each prediction step parameter values of the devices on the level K are defined using the set of rules R. Further, parameter values for levels $(0 \dots K - 1)$ are calculated by aggregation of parameter values

according to the hierarchy. In this way, additionally $\sum_i q_i$ operations of values aggregation should be done. Let $P^l_{t,i}$ is the incoming traffic for i-th model element $\widetilde{D}_{t,i}$ on level l. This model element is hierarchically connected to elements on level $l-1$. In this case traffic for element $\widetilde{D}_{t,i}$ can be calculated as: $P^l_{t,i} = \sum_{m=1}^{q_{l-1}} P^{l-1}_{t,m}$.

For the considered situation, when the values of the parameters are predicted only for the elements of the lowest level K, we need to consider all elements and links defined in the model to calculate parameters values of the upper level elements.

It is not the only case when the rules are defined only for the lowest level elements. Commonly the rules are also known for the elements on levels $(0 \ldots K-1)$, then the state of the elements of upper levels are estimated using the rules defined for the corresponding level. This is especially useful when analysis on lower levels is not requested.

D. *General procedure for prediction task solving*

Suggested procedure for networks state prediction based on knowledge graphs is shown in Fig. 2.

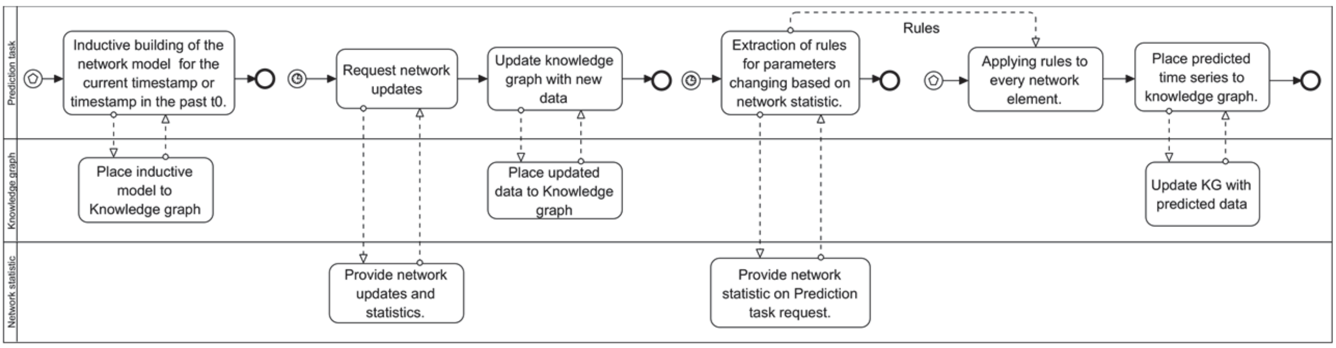

Fig. 2 Prediction procedure general view

The procedure consists of following steps:

- **Inductive building of the network model based on knowledge graph for the current timestamp or timestamp in the past t_0.** The input data for the inductive building of multi-level TN knowledge graphs are static TN models, and statistical data. The output is a synthesized multi-level knowledge graph. Knowledge graph is enriched with data about network state for the timestamp t_0. Feed of this data are network monitoring systems, management systems, billing systems etc. The examples of inductive models based on knowledge graphs have been studied in the articles [14, 15, 16].
- **Extraction of rules of parameters changing based on network statistic.** The rule R_j for model element D_i is used for transforming the model of the element $D_{t,i}$ to $D_{t+t_d,i}$. To create the set of rules, available statistical data is analyzed, timestamps and conditions of transformations are defined. System elements log analysis and network analytical systems can be used for rules extraction [18].
- **Apply rules to the models of the network element. At** every prediction timestamp t_m, the following operations are done:
  - Every model element D_i is analyzed and is checked if any rule R_j can be used for element transforming (check conditions).
  - The rules R_j that can be used in the observed conditions are applied. This process results in building the sequence of models for each element i: $\{D_{1,i}, D_{2,i} \ldots, D_{m,i}\}$.
- **Place predicted values of the parameters of the elements $\{D_1, D_2 \ldots, D_m\}$ to knowledge graph.**

III. MODEL BUILDING ALGORITHM

High level algorithm activity diagram is shown in Fig. 3.

Inductive synthesis is used for building TN model for the current timestamp t_0. Then prediction rules are extracted from network statistic and additional prior information like services and network events schedules. Using prediction rules, the algorithm predicts the state of every element on each prediction step.

The task is solved for network state prediction for timestamps $t_1, t_2, \ldots, t_w$ (or in terms of steps - $1,2,\ldots,m$) within the following conditions:

- The model hierarchy structure for timestamps $t_1, t_2, \ldots, t_w$ is the same as for t_0;
- Every model element is characterized by parameters values. The possible changes of the parameters values at different timestamps are defined by the corresponding rules.

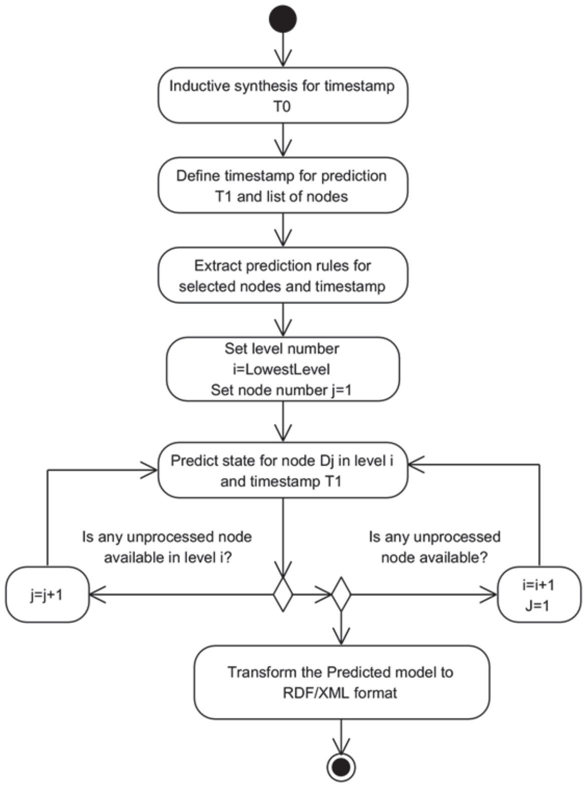

Fig. 3 Model synthesis activity diagram

The "Model synthesis" algorithm is following:

Algorithm 1 Model synthesis algorithm

//Inductive synthesis for timestamp t_0
Create the Inductive model - I // Inductive synthesis algorithm should be used

Set t_0 as current or past timestamp
Set t_d //Prediction step

// Define timestamp t_w and list of model elements for prediction
Set $t_w =$ user defined value //Set t_w as user defined timestamp

Set $U =$ user defined value // Set the set of model nodes for prediction

//Find the user model subset $D_{t_0,n}$ for predicted model building as intersection of sets: I and U for initial timestamp t_0.
$D_{t_0,n} = I \bigcap U$

//Extract prediction rules for selected nodes
// R- the set of rules of changing predicted parameters values. The rule R_i for the node $\widetilde{D}_{t,i}$, where t – timestamp identifier

//Using rules for predicting model state
Set $t = t_0$
***while** ($t \le t_w$) {*
Set $i = K$ //LowestLevel
***while**($i \ge 0$) {*
***for** (every $D_{t,n}$ where level number $= i$) {*
***for** (each rule R_l) {*
***if**(R_l is applicable for $D_{t,n}$) {*
Set $D_{t+t_d,n} = R_l(D_{t,n})$
}
}
Set $i = i - 1$ //Change model level
}
Set $t = t + t_d$ //Move to next step
}

// Transform the Predicted model to RDF/XML format
***for** (each $D_{t_w,n}$) {*
Convert entry $D_{t_w,n}$ to set of RDF triple form
}

End

The suggested approach can be used for solving cable TV operators tasks. Incoming internet traffic can be of interest in network performance prediction. In this case prediction model of changing incoming traffic should be built for all user devices. The prediction rules are commonly defined on the base of statistics of user preferences, time, day of week, EPG (Electronic Program Guide) data. The predicted data is retrieved from RDF storage using SPARQL requests. The following request can be used for data retrieving:

- Get parameter values for the selected element.
- Get parameter values for the selected level.
- Get time series of parameters values for the selected element/level.

E. Prediction accuracy analysis

Accuracy of the suggested method is analyzed based on Cable TV operator model using generic data. The following parameters have been used:

- Network type: QAM Television network (DOCSIS internet protocol supported).
- Type of analyzed traffic: Internet traffic.
- Network structure – hierarchical topology, within one regional hub.
- Existing static models: Network topology, Billing model, Services model, Available information resources.
- Number of model elements: 1M (about 500k users and 500k user devices).
- Used rules: traffic distribution by hours in a day, days in a week, days in a month; user preferences, user age.
- Models were built for prediction within a day, day in week, day in month.
- Results were calculated for every node in topology model from user devices to regional hub.

The results are shown in Table I.

TABLE I. PREDICTION ACCURACY ANALYSIS

#	Experiment description	Accuracy, %
1	Traffic prediction for hour within one day	95,12
2	Traffic prediction for day within one week	97,55
3	Traffic prediction for day within one month	96,89

Analysis confirms that accuracy of prediction is not less than 95%.

IV. CASE STUDY

One of the actual tasks of telecommunication network operators is to control the bandwidth and performance of data centers and channels due to their capacity should be planned before. It is very often required to have information in advance about the needs of subscribers for network services and consequently, in requested network traffic. For solving this problem, we can build a model of traffic consumption by subscriber devices. We take rules of the consumption of network traffic depending on various factors from the network statistics. Analyzing the data of user devices at different network levels, we can analyze the state of the whole network. Inductive method is used for building the model for timestamp t_0. To solve the problem of predicting state in the future, the suggested method can be used. The predicted data is represented in the form of a knowledge graph and the results of the prediction data are retrieved using SPARQL queries.

A. The task

We predict the load of the set of telecommunication network nodes for Cable TV operator network. The structure of TN is the same as was discussed in [15]:

- Billing model.
- Access rights model (lists of access rights in the perspective of network users).
- Network topology model.
- Network applications hierarchy model.
- Network service hierarchy model.
- Data model.

According to the extracted rules traffic depends on the following factors:

- Daily schedule of available services.
- Working day / weekend / holidays.
- Subscribers age.

We define traffic based on prediction of using services on user devices. Using of service means using traffic. For this task we limit the number of parallel working services by 4.

B. Knowledge graph synthesis

The knowledge graph for the task solving contains nodes that reflect the Network hierarchy (Fig. 4) and the Services hierarchy (Fig. 5). The Network hierarchy describes the devices that are linked to each other and to user devices (STBs). The Services hierarchy defines the set of available telecommunication services. Using of services creates STB traffic load. In the case study we analyze the network with the following parameters:

- Number of network levels – 4 (0 – 3).
- Number of network devices – 7.
- Number of user devices – 9.
- Number of services – 4 (Watch TV, VOD, DVR, PPV)

The knowledge graph dynamic structure for prediction parameter values is presented in Fig. 6.

C. Rules extraction and using

We extract rules only for user devices, traffic for network devices is aggregated according to the hierarchy structure (Fig.3). The data is available for rules extraction. The fragment of data for one user device for one service is presented below. The column ***INTERVAL_ID*** defines time intervals withing a day. Column ***SERVICE*** defines the used service, column ***PROBABILITY*** defines probability of using a service (ON condition of service).

The fragment of rules defined for User devices:

```
INTERVAL_ID,SERVICE,PROBABILITY
00,VOD,0.004
00,PPV,0.5
01,VOD,0.004
01,WatchTV,0.6
…
```

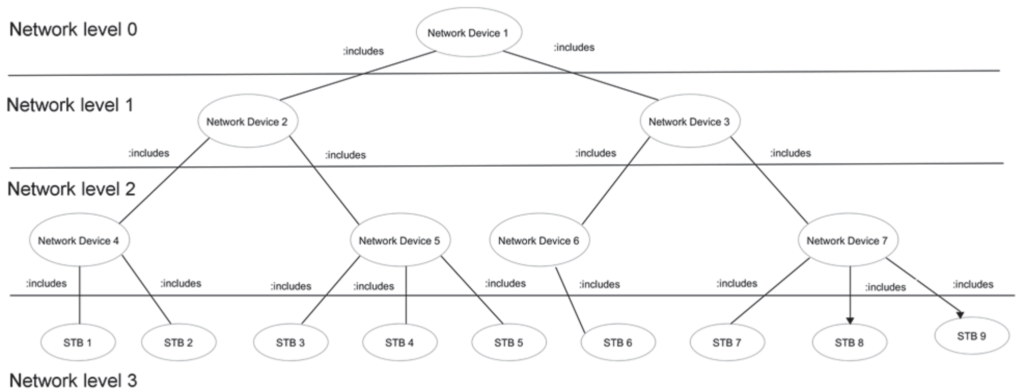

Fig. 4 Knowledge graph network structure

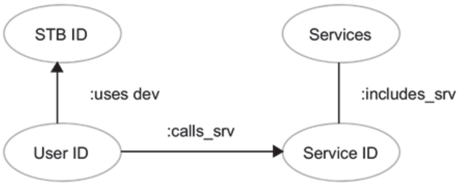

Fig. 5 Knowledge graph services structure

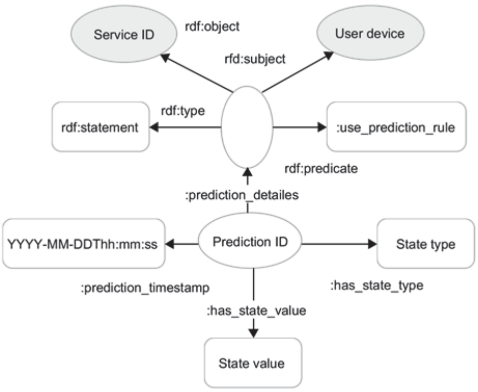

Fig. 6 Knowledge graph dynamic structure for parameter values prediction

D. *Results retrieved and discission*

Special program for model creation and traffic prediction was developed. The program code in Python language and result knowledge graph in RDF/XML format are available in GitHub repository [17].

Predicted data can be retrieved by the following SPARQL requests:

Request #1: Get parameter values for selected element (User_device_1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
 ?Prediction_ID my:prediction_timestamp "2020-11-19T04:00:00" .
 ?Prediction_ID my:has_state_value ?Value .
 ?Prediction_ID my:prediction_detailes ?Prediction_detailes .
 ?Prediction_detailes rdf:subject <http://127.0.0.1/STB_1/> .
 ?Prediction_detailes rdf:object ?Service .
}
```

Fragment of response:

Prediction_ID	Value	Prediction_detailes	Service
<http://127.0.0.1/Prediction_113/>	OFF	t1142	WatchTV
<http://127.0.0.1/Prediction_112/>	ON	t1285	VOD
<http://127.0.0.1/Prediction_114/>	ON	t1363	nPVR
<http://127.0.0.1/Prediction_115/>	ON	t1400	PPV

Request #2: Get parameter values for selected level (1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
 {<http://127.0.0.1/NW_Device_2/> my:includes ?Devices_2 .
 ?Devices_2 my:includes ?Devices_3 .
 ?Prediction_node rdf:subject ?Devices_3 .
 ?Prediction_node rdf:object ?Service .
 ?Prediction_ID my:prediction_detailes ?Prediction_node .
 ?Prediction_ID my:has_state_value ?Value .
 ?Prediction_ID my:prediction_timestamp "2020-11-19T01:00:00" .}
 UNION
 {<http://127.0.0.1/NW_Device_3/> my:includes ?Devices_2 .
 ?Devices_2 my:includes ?Devices_3 .
 ?Prediction_node rdf:subject ?Devices_3 .
 ?Prediction_node rdf:object ?Service .
 ?Prediction_ID my:prediction_detailes ?Prediction_node .
 ?Prediction_ID my:has_state_value ?Value .
 ?Prediction_ID my:prediction_timestamp "2020-11-19T01:00:00" .}
}
```

Fragment of response:

Devices_2	Devices_3	Prediction_node	Service	Prediction_ID	Value
<http://127.0.0.1/NW_Device_4/>	<http://127.0.0.1/STB_1/>	t1334	VOD	<http://127.0.0.1/Prediction_100/>	ON
<http://127.0.0.1/NW_Device_4/>	<http://127.0.0.1/STB_1/>	t1604	WatchTV	<http://127.0.0.1/Prediction_101/>	ON
<http://127.0.0.1/NW_Device_4/>	<http://127.0.0.1/STB_1/>	t1621	nPVR	<http://127.0.0.1/Prediction_102/>	OFF
<http://127.0.0.1/NW_Device_4/>	<http://127.0.0.1/STB_1/>	t1455	PPV	<http://127.0.0.1/Prediction_103/>	OFF

…

Request #3: Get time series of parameter values for selected element (User_device_1)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
  ?Prediction_detailes rdf:subject <http://127.0.0.1/STB_1/> .
  ?Prediction_ID my:prediction_detailes ?Prediction_detailes .
  ?Prediction_ID my:has_state_value ?State_value .
  ?Prediction_ID my:prediction_timestamp ?Time .
 }
```

Fragment of response:

Prediction_detailes	Prediction_ID	State_value	Time
t1334	<http://127.0.0.1/Prediction_100/>	ON	2020-11-19T01:00:00
t1604	<http://127.0.0.1/Prediction_101/>	ON	2020-11-19T01:00:00
t1621	<http://127.0.0.1/Prediction_102/>	OFF	2020-11-19T01:00:00

…

V. Conclusion

The new method of prediction the state of telecommunication networks is suggested in the paper. The discussed method predicts the state of a network through predicting parameters values of its elements. The method is based on using knowledge graphs, which allows integrate static models of telecommunication networks, statistical and operational data within a single model. Prediction accuracy of the proposed method is not less than prediction accuracy of used rules for model elements state prediction. Using of hierarchical models for network structure, it is possible to predict or calculate parameter values for different levels according to the user request. This option allows to reduce computational complexity of prediction if only high-level elements states are requested for prediction. The case study of internet traffic prediction considered in the paper reflects how suggested approach can be used for particular Cable TV operator task solving. In the course of further research prediction of network state when various factors influence the network state in parallel and in opposite ways will be considered.

References

[1] Joshi, Manish & Aldhayni, TheyaznTheyazn. (2015). A Review of Network Traffic Analysis and Prediction Techniques.

[2] N. K. Hoong, P. K. Hoong, I. K.T. Tan, N.M Muthuvelu, L. C. Seng, Impact of Utilizing Forecasted Network Traffic for Data Transfers) IEEE, 211) 978-1-4244-8830-8.

[3] Y.Yu, M. Song, Z. Ren, l. Song, Network Traffic Analysis and Prediction Based on APM (IEEE, 2011) 978-1-4577-0208-2/11.

[4] 20. N.Sadek, A. Khotanzad, Multi-scale High Speed Network Traffic Prediction Using K-Factor Gengendaue ARMA Model (IEEE, 2004) 2148-2152.

[5] C Park, D-M Woo, Prediction of Network Traffic by Using Dynamic Bilinear Recurrent Neural Network)IEEE, 2009) 978-0-7695-3736-8.

[6] N.CAnand,C. ScoglioS, B. Natarajan, GARCH Non-Linear Time Series Model for Traffic Modeling and Prediction (IEEE, 2008) 978-1-4244-2065-0.

[7] E. S. Yu, C.Y.R.Chen, Traffic Prediction Using Neural Networks (IEEE, 1993) 0-7803-0917-0.

[8] S.M.A Burney, S. Raza, Monte Carlo Simulation and Prediction of internet load using conditional mean and conditional variance model (ICCS-IX, 2007).

[9] D. Zeng, J. Xu1, J. Gu , L.Liu , G. Xu, Short Term Traffic Flow Prediction Using Hybrid ARIMA and ANN model (IEEE, 2008) 978-0-7695-3342-1.

[10] B. Zhou,D. He, Z. Sun, Traffic predictable based on ARIMA/GARCH Model (IEEE, 2006) 0-7803-9455-0.

[11] C.Guang,G.Jian,D.Wei, A Time series Decomposed Model of Network Traffic(Springer, 2005) 338-345.

[12] Resource Description Framework (RDF) Model and Syntax Specification: URL: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[13] Basu Sabyasachi, Mukherjee Amarnath, Klivansky Steven M., Time Series Models for Internet Traffic (2005), http://hdl.handle.net/1853/6696

[14] Kulikov I., Wohlgenannt G., Shichkina Y., Zhukova N. (2020) An Analytical Computing Infrastructure for Monitoring Dynamic Networks Based on Knowledge Graphs. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science, vol 12254. Springer, Cham. https://doi.org/10.1007/978-3-030-58817-5_15

[15] K. Krinkin, I. Kulikov, A. Vodyaho and N. Zhukova, "Architecture of a Telecommunications Network Monitoring System Based on a Knowledge Graph," 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 2020, pp. 231-239, doi: 10.23919/FRUCT48808.2020.9087429

[16] K. Krinkin, A. Vodyaho, I. Kulikov and N. Zhukova, "Models of Telecommunications Network Monitoring Based on Knowledge Graphs," 2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2020, pp. 1-7, doi: 10.1109/MECO49872.2020.9134148

[17] GitHub project. https://github.com/kulikovia/FRUCT-2021

[18] Anton Chuvakin, Kevin Schmidt, Chris Phillips, Chapter 15 - Tools for Log Analysis and Collection, Editor(s): Anton Chuvakin, Kevin Schmidt, Chris Phillips, Logging and Log Management, Syngress, 2013, Pages 243-266, ISBN 9781597496353, https://doi.org/10.1016/B978-1-59-749635-3.00015-4. (http://www.sciencedirect.com/science/article/pii/B9781597496353000154)