International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Series Y

Supplement 70
(07/2021)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

ITU-T Y.3800-series – Quantum key distribution networks – Applications of machine learning

ITU-T Y-series Recommendations – Supplement 70

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Supplement 70 to ITU-T Y-series Recommendations

ITU-T Y.3800-series – Quantum key distribution networks – Applications of machine learning

Summary

For quantum key distribution networks (QKDNs), Supplement 70 to ITU-T Y-series Recommendations presents the applications of machine learning (ML) in the quantum layer, the key management layer and the management and control layers of QKDN including the use case background, issues, role of ML in QKDN, use case analysis, as well as benefits and impact.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y Suppl. 70	2021-07-16	13	11.1002/1000/14757

Keywords

Applications, machine learning (ML), quantum key distribution (QKD), QKD networks.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the publication development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Supplement 70 to ITU-T Y.3800-series Recommendations

ITU-T Y.3800-series – Quantum key distribution networks – Applications of machine learning

1 Scope

This Supplement presents the applications of machine learning (ML) in quantum key distribution networks (QKDNs).

In particular, this Supplement includes:

– overview of ML applications in QKDN;

– applications of ML in the quantum layer of QKDN;

– applications of ML in the key management layer of QKDN;

– applications of ML in the control and management layers of QKDN.

2 References

[ITU-T Y.3170] Recommendation ITU-T Y.3170 (2018), *Requirements for machine learning-based quality of service assurance for the IMT-2020 network.*

[ITU-T Y.3172] Recommendation ITU-T Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020.*

[ITU-T Y.3800] Recommendation ITU-T Y.3800 (2019), *Overview on networks supporting quantum key distribution*.

[ITU-T Y.3801] Recommendation ITU-T Y.3801 (2020), *Functional requirements for quantum key distribution networks*.

[ITU-T Y.3802] Recommendation ITU-T Y.3802 (2020), *Quantum key distribution networks – Functional architecture.*

[ITU-T Y.3803] Recommendation ITU-T Y.3803 (2020), *Quantum key distribution networks – Key management*.

[ITU-T Y.3804] Recommendation ITU-T Y.3804 (2020), *Quantum key distribution networks – Control and management*.

3 Terms and definitions

3.1 Terms defined elsewhere

This Supplement uses the following terms defined elsewhere:

3.1.1 machine learning (ML) [ITU-T Y.3172]: Processes that enable computational systems to understand data and gain knowledge from it without necessarily being explicitly programmed.

NOTE 1 – This definition adapted from [b-ETSI GR ENI 004].

NOTE 2 – Supervised machine learning and unsupervised machine learning are two examples of machine learning types.

3.1.2 machine learning model [ITU-T Y.3172]: Model created by applying machine learning techniques to data to learn from.

NOTE 1 – A machine learning model is used to generate predictions (e.g., regression, classification, clustering) on new (untrained) data.

NOTE 2 – A machine learning model may be encapsulated in a deployable fashion in the form of a software (e.g., virtual machine, container) or hardware component (e.g., IoT device).

NOTE 3 – Machine learning techniques include learning algorithms (e.g., learning the function that maps input data attributes to output data).

3.1.3 machine learning output [b-ITU-T Y Suppl. 55]: Policies or configurations to be applied in the network, based on the output from the machine learning model.

NOTE – The target of machine learning output may be functions in the network.

3.1.4 machine learning pipeline [ITU-T Y.3172]: A set of logical nodes, each with specific functionalities, that can be combined to form a machine learning application in a telecommunication network.

NOTE – The nodes are entities that are managed in a standard manner and can be hosted in a variety of network functions.

3.1.5 quantum key distribution [b-ETSI GR QKD 007]: Procedure or method for generating and distributing symmetrical cryptographic keys with information theoretical security based on quantum information theory.

3.1.6 quantum key distribution network (QKDN) [ITU-T Y.3800]: A network comprised of two or more quantum key distribution (QKD) nodes connected through QKD links.

NOTE – A QKDN allows sharing keys between the QKD nodes by key relay when they are not directly connected by a QKD link.

3.1.7 quantum key distribution node [ITU-T Y.3800]: A node that contains one or more quantum key distribution (QKD) modules protected against intrusion and attacks by unauthorized parties.

NOTE – A QKD node can contain a key manager (KM).

3.2 Terms defined in this Supplement

None.

4 Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

AES Advanced Encryption Standard

ANN Artificial Neural Network

C Collector

D Distributor

ID Identifier

KM Key Manager

KMA Key Management Agent

KSA Key Supply Agent

LSTM Long Short-Term Memory

M Model

ML Machine Learning

P Policy

PP Pre-Processor

OSNR Optical Signal-to-Noise Ratio

QBER Quantum Bit-Error Ratio

QKD Quantum Key Distribution

QKDN Quantum Key Distribution Network

RNN Recurrent Neural Network

RUL Remaining Use Life

SPD Single Photon Detector

SRC Source

5 Conventions

None.

6 Overview

QKDN is a technology that extends the reachability and availability of QKD, which is stated in [ITU-T Y.3800]. It is comprised of two or more QKD nodes connected through QKD links. The main goal of a QKDN is to increase the security of key distribution. In a QKDN, two or more designated parties in a user network can share the keys for various cryptographic applications. However, the challenge of operating a QKDN efficiently increases with the scale of the network.

ML mechanisms are able to teach a computer to learn knowledge using data without being explicitly programmed. ML can be applied to the networking field, which can intelligently learn the network environment and react to dynamic situations [ITU-T Y.3170]. There are some applications of ML in telecommunication networks, such as prediction of traffic and faults. There is increasing interest in and necessity for applying ML to improve QKDN performances. Due to the advantages of ML, ML can be applied in QKDNs so as to improve QKD performance and the control and management efficiency of a QKDN.

Figure 6-1 shows an ML pipeline subsystem in a QKDN. The ML functions support a set of functional elements in an ML pipeline subsystem including collector (C), pre-processor (PP), model (M), policy (P) and distributor (D). The ML functions are able to collect input data from a source (SRC) of data through data handling interfaces. The SRC can be in different layers of QKDNs. The target of the ML output (SINK) can be elements in the quantum, key management, as well as QKDN control and management layers. More details related to ML pipeline subsystems can be found in [ITU-T Y.3172]. The SRC can be in different layers of QKDNs [ITU-T Y.3802].

In clauses 7 to 9, based on academic and industrial advances, this Supplement presents several applications of ML in the quantum, key management, as well as the management and control layers of QKDN including the use case background, issues, role of ML in QKDN, use case analysis, in addition to benefits and impact.

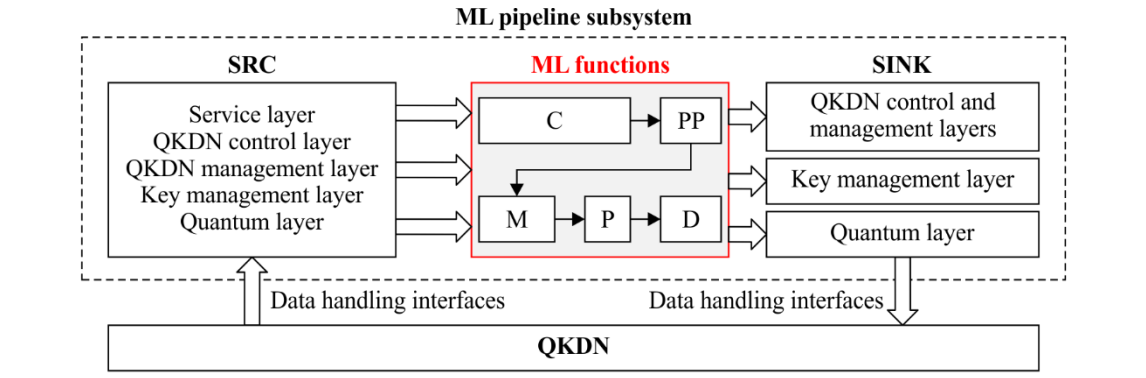

Figure 6-1 – ML pipeline subsystem in a QKDN

7 Application of ML to the quantum layer of a QKDN

7.1 Introduction

ML is applied to the quantum layer of a QKDN to improve performance. Three use cases are presented including ML-based quantum channel performance prediction, ML-based QKD system parameter optimization and ML-based remaining use life (RUL) prediction of components in a QKD system.

7.2 Use case QL01: ML-based quantum channel performance prediction

Use case description

(1) Background

Relatively stable and predictable quantum channel performance and transmission quality in the quantum layer is crucial for the implementation and commercialization of QKDNs. The main challenge is that the noise falls into the quantum channel, thereby reducing the quality of quantum channel and causing low key rate, especially when quantum-encoded photons coexist with high-intensity classical signals. Recently, ML-based techniques have been applied to optical communication to predict its optical signal-to-noise ratio (OSNR).

(2) Issues

QKD is the most widely researched area in quantum communication projects to achieve secure transmission. However, low key rate is a significant challenge for practical QKD. Key rate is often related to parameters such as single photon detector (SPD), photon detection output count and quantum bit-error ratio (QBER). It changes with the attenuation of the quantum channel. QBER has become one of the most crucial monitoring parameters. To avoid the drop in key rate caused by channel noise, it is helpful to apply ML-based techniques to predict quantum channel performance.

(3) Role of ML in QKDN

During the QKD process, noise in the quantum channel causes the quantum channel quality and generated key rate to deteriorate. Figure 7-1 is a schematic diagram of ML-based quantum channel performance prediction. First, ML functions collect quantum channel-related data and record the corresponding quantum channel performance through quantum channel measurement for ML model training and testing. With the trained ML model, quantum channel performance can then be predicted based on the current input quantum channel-related data. Second, according to the predicted channel performance, feedback can be adjusted in advance to improve the channel environment and reduce unnecessary loss caused by key rate decreases.

A supervised ML method can be beneficial to estimate quantum channel performance (noise, QBER, etc.) when various quantum channels, allocated spectrum, launch power and channel spacing exist.

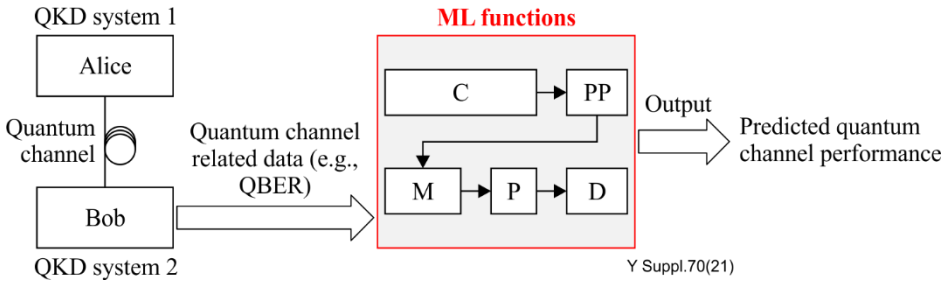

Figure 7-1 – ML-based quantum channel performance prediction

Use case analysis

– Analysis related to data collection follows.

1) The ML functions collect quantum channel parameters through quantum channel measurement under the influence of different noise environments.
2) The collected data includes the quantum channel performance-related parameters (e.g., QBER of quantum channel, the SPD output counter, code formation rates under different noise environments).

– Analysis related to data storage and processing follows.

1) ML-based quantum channel performance prediction supports the storage of data used for analytics. A database in the quantum layer stores the collected data and possibly stores predictions.
2) It supports the pre-processing and intelligent analysis of the input data.

– Analysis related to application of ML output, as follows.

1) The ML output is applied to predict quantum channel performance based on input quantum channel parameters.
2) The quantum layer can be configured according to the ML output to ensure robust network operations.

Benefits and impact

The ML-based quantum channel performance prediction method forecasts quantum channel performance under different channel noise environments. Measures can be taken in advance based on the predictions to improve the channel environment and put the quantum channel in an optimal performance state.

7.3 Use case QL02: ML-based QKD system parameter optimization

Use case description

(1) Background

In a practical QKD system, the low efficiency of basis-sift factors, the selection of intensities and the probability of sending the selected intensities are essential to obtain the optimal quantum key rate within a certain service delivery time. It is important to maintain a QKD system at best performance and necessary to optimize the parameters of the QKD system. QKD system parameters include; the intensities of signal and decoy state; the probabilities to choose different intensities and bases; as well as the probability that QKD receiver Bob measures the incoming pulse on the *Z* basis, etc.

(2) Issue

Practically, the insufficient computing power of a QKD system either causes a lag time for an optimization off-line (and hence delay) or uses suboptimal or even unoptimized parameters in real time, which reduces the efficiency of the basis-sift factor. At the same time, when the gain and QBER change with the environment, the QKD system parameters also need to be re-optimized. It is a difficult task to optimize the parameters of QKD system quickly and accurately. ML algorithms can help by learning from a large number of training data, which will efficiently optimize the parameters of a QKD system.

(3) Role of ML in QKDN

The ML-based QKD system parameter optimization solution pre-executes the optimization algorithm before key generation. Figure 7-2 is a schematic diagram of ML-based QKD system parameter optimization. First, the input data is sampled to pick a random combination of physical parameters that cannot be controlled by users, and use a local search algorithm to calculate their corresponding optimization parameter values that can be adjusted by users. The physical parameter values obtained are input into the ML model trainer of the ML model selected. After training, *N* sets of prediction parameter values are output. The result of the comparison of the key rate obtained by the classical algorithm with that based on predicted parameter values is fed back into the ML model trainer. Second, when the QKD system needs parameter optimization, real-time data is input and the optimal parameter values are output after applying the ML functions. Last, the configuration parameters are input into the QKD system to complete parameter optimization.

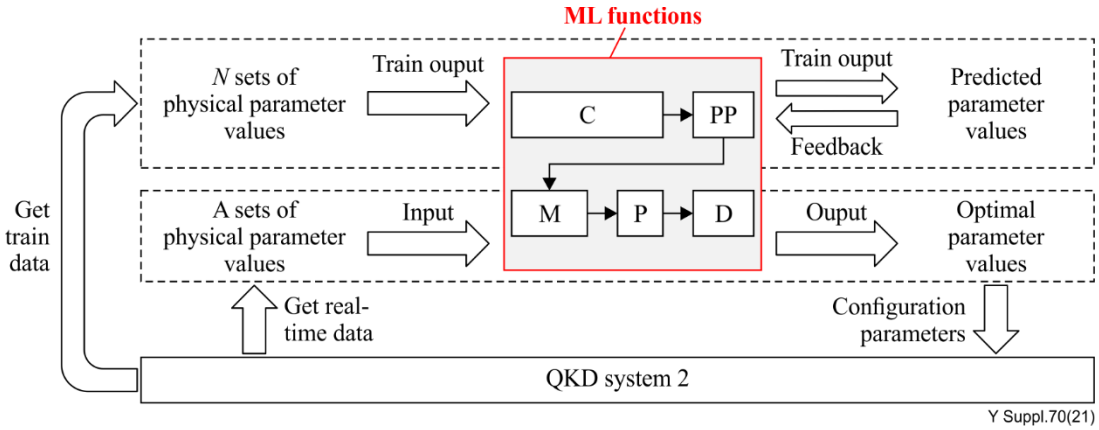

Figure 7-2 – ML-based QKD system parameter optimization

Use case analysis

– Analysis related to data collection follows.

1) An ML-based QKD system randomly samples the input data to pick a random combination of physical parameters and uses a local search algorithm to calculate the corresponding optimization parameters.
2) It collects the optimal parameters (e.g., the choice of signal intensities and the probabilities of sending them) by classical algorithms in the quantum layer.
3) It collects the physical parameters (e.g., distance between two QKD users, the detector efficiency, the dark count probability, the basis misalignment, the error-correction efficiency and number of signals) in the quantum layer.

– Analysis related to data storage and processing follows.

1) An ML-based QKD system supports simple scaling and normalization of input data.
2) It supports real-time prediction that aims to forecast optimal parameters.
3) It supports retraining the ML model after the update of datasets.

– Analysis related to application of ML output follows.

1) The ML output is compared with previous parameter settings.
2) The ML output is applied before key generation.

Benefits and impact

The ML-based QKD system parameter optimization solution operates quickly and accurately based on the real-time changing environment, maintaining the QKD system in the optimal performance state in real time.

7.4 Use case QL03: ML-based RUL prediction of components in a QKD system

Use case description

(1) Background

The life cycle of components in a QKD system is essential for normal operation. Extension in the working hours of components induces aging, which will cause component life to end suddenly, greatly influencing key generation and key supply. ML can build a training model to predict the RUL of components by collecting relevant data, including the working time of components, the operating conditions and component life cycle data.

(2) Issues

Considering the influence of external environments and internal factors, it is not known when components will stop working. The components of a QKD system include a pulsed light source, decoy state modulation module, random number generator and SPD. It is important to make an accurate prediction of the RUL of components in QKD systems. The ML technique has the ability to predict the RUL of components by using a large number of relevant data.

(3) Role of ML in QKDN

Figure 7-3 is a schematic diagram of ML-based RUL prediction of components in a QKD system. First, component parameters in a QKD system, such as temperature, power and voltage, are collected, classified and pre-processed. The pre-processed data is then input into the ML functions. ML model training produces ML output. Finally, the predicted RUL is output. Considering the long life of lasers and the prediction target, long short-term memory (LSTM) is suggested to predict laser RUL.

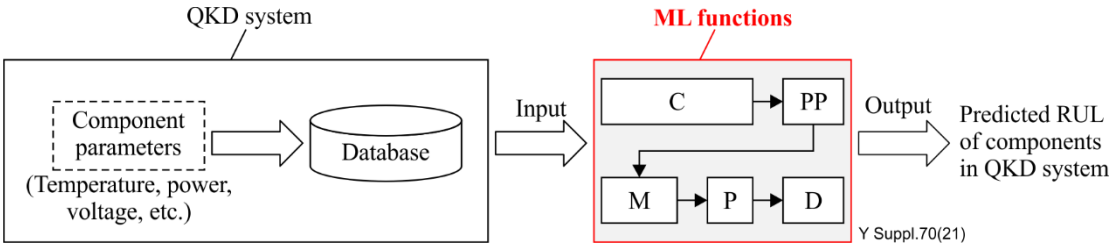

Figure 7-3 – ML-based RUL prediction of components in a QKD system

Use case analysis

– Analysis related to data collection follows.

1) ML-based RUL prediction of components collects different parameters (e.g., temperature, power, voltage) values of components in a QKD system.
2) It collects relevant historical data for ML model training.
3) It collects the relevant values periodically using different sensors.

– Analysis related to data storage and processing follows.

1) It supports storage of the measured values used for analysis.

2) It supports data processing, such as normalization and segmentation.

– Analysis related to application of ML output follows.

1) The ML output is applied to assess the component status and RUL.

Benefits and impact

ML-based RUL prediction of components in a QKD system solution helps to ensure stable QKD system operations.

8 Applications of ML in the key management layer of a QKDN

8.1 Introduction

The applications of ML in the key management layer of a QKDN improve key management efficiency and stability. Three use cases are presented including ML-based key formatting, ML-based key storage management, and ML-based suspicious behaviour detection in the key management layer.

8.2 Use case KM01: ML-based key formatting

Use case description

(1) Background

Keys, on supply or relay, need to be combined or split if their lengths are not appropriate, as stated in [ITU-T Y.3803]. There are different key formats for various security requirements of services and different encryption algorithms (e.g., one time password, advanced encryption standard-512 (AES-512), AES-256, AES-128). To maintain interconnectivity and expandability in the QKDN, an appropriate key format needs to be solved for key data with added metadata containing various types of information.

(2) Issues

Keys require formatting before storage. The lengths of the acquired QKD-key files may differ from each other. As recommended by the Req_KM 3 of [ITU-T Y.3801], the KM agent re-formats (combines or splits) the QKD keys into a prescribed unit length, and then temporarily stores them in a buffer for further key supply.

In QKDNs, different services with dynamic arrival and various types might request different numbers of keys with varying formats. A QKDN may need to re-format keys before their supply, which will introduce time cost and risk of key non-synchronization failure. The ML technique has the ability to find rules from a large amount of service data and predict future service characteristics. Compared with applying traditional methods, such as expert systems, the quality of the predicted statistical service characteristics by applying ML is improved, e.g., in terms of prediction accuracy.

(3) Role of ML in QKDN

The ML-based key formatting solution operates before key storage with awareness of service characteristics. Figure 8-1 is a schematic diagram of ML-based key formatting. A large amount of service information during a certain period is input into the ML functions for training. The output of the ML functions is the predicted service characteristics. According to the output, KM formats and then stores keys in the store for future supply. Since the number of stored keys with definite formats is based on service characteristics, the times of key re-formatting will reduce while supplying keys for services. As for the ML models, prediction models, such as deep learning algorithms and the Elman neural network, can be applied.

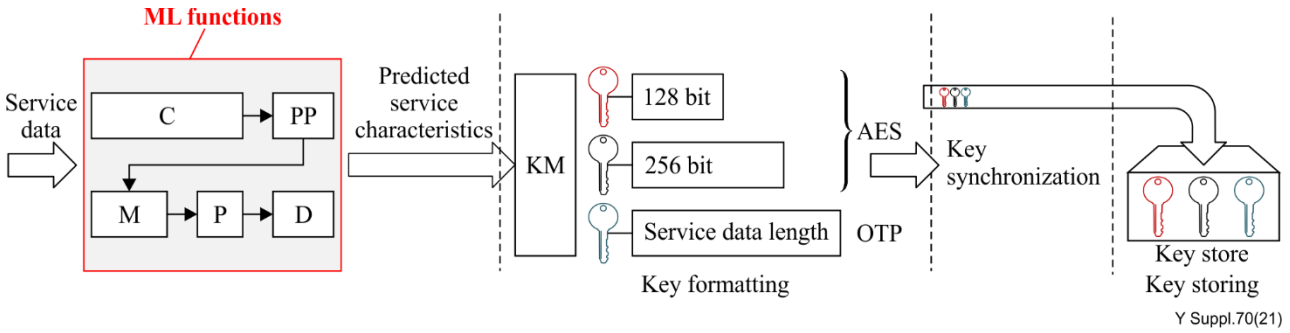

Figure 8-1 – ML-based key formatting

Use case analysis

– Analysis related to data collection follows.

1) ML-based key formatting collects service data from the service layer. The service data is collected continuously for updating ML model in order to improve the accuracy and effectiveness of the ML model in real time.

2) The collected service data includes the service characteristics (e.g., service arrival time, service duration time, service security levels and the size of service data that needs encryption with keys). Note that the service security level is used to describe the security requirement of a service.

– Analysis related to data storage and processing follows.

1) It supports storage of data used for analysis.

2) It supports real-time prediction that aims to forecast service characteristics.

3) It supports predictions at different time granularity, such as real-time predictions (user activity), short-term predictions (user group activity) and long-term predictions (large-scale activity).

– Analysis related to application of ML output follows.

1) The ML output is applied in formatting keys before their storage.

2) The KM is able to format keys according to the ML output.

Benefits and impact

The ML-based key formatting solution operates with the awareness of service characteristics before storing keys, which reduces time cost and the risk of key non-synchronization failure during key supply.

8.3 Use case KM02: ML-based key storage management

Use case description

(1) Background

Since the services are dynamic and extensive, it is necessary to have efficient key storage management, so as to realize the reasonable scheduling and efficient utilization of key resources. In the key management layer, KM is responsible for receiving and managing keys generated by QKD modules, relaying keys under the control of the QKDN controller, and providing keys to the service layer, as stated in [ITU-T Y.3800].

(2) Issues

The key requirement of services changes dynamically in the actual situation. On one hand, services may not be supplied with keys successfully because of many factors including insufficient key resources in key storage and keys with long storage time. On the other, when the key requirement is

greatly reduced, many keys in key storage will not be used, which leads to unnecessary redundancy of keys. Hence, it is important to evaluate the health state of key storage. However, it is difficult for existing traditional solutions to accurately perceive the actual needs of services and establish an evaluation scheme. The ML technique is a strong tool to find rules from a large amount of data.

(3) Role of ML in QKDN

The ML-based key storage management solution reasonably evaluates and predicts the health state of key storage. Figure 8-2 is a schematic diagram of ML-based key storage management. ML functions collect a lot of training data, use ML algorithms to build an ML model, and constantly adjust it through result comparison, so as to accurately evaluate and predict health states of key storage. KM manages keys in storage based on the output value of ML functions, so as to ensure reasonable scheduling and efficient utilization of key resources, as well as avoiding problems such as long key storage time or excessive jitter. The key management agent (KMA) manages the key life cycle, which archives or destroys keys that have been stored for a long time. Finally, keys are supplied to cryptographic applications on demand if key storage is in a health state.

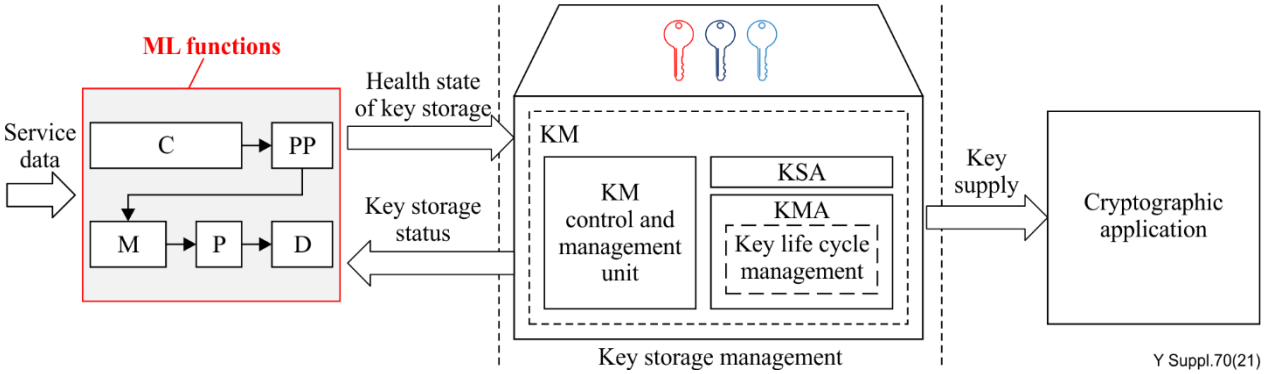

Figure 8-2 – ML-based key storage management

Use case analysis

– Analysis related to data collection follows.

1) ML-based key storage management collects service data from the service layer (e.g., service type, security level, required key quantity) in real time.
2) It collects key storage status (e.g., key numbers, key life cycle).

– Analysis related to data storage and processing follows.

1) It supports the storage of a large amount of training data.
2) It supports the retraining of the ML model after data updates.

– Analysis related to application of ML output follows.

1) The ML output is applied in KMA to control the key life cycle.
2) The ML output is applied in the KM control and management unit to feedback the state of key generation to the QKDN controller.

Benefits and impact

An ML-based key storage management solution evaluates and predicts the health state of key storage and helps efficient utilization of key resources.

8.4 Use case KM03: ML-based suspicious behaviour detection in the key management layer

Use case description

(1) Background

Suspicious behaviour detection exists in the quantum and key management layers, and this use case focuses on the second scenario. The key supply agent (KSA) authenticates the cryptographic application by an appropriate method. Their certificate can be issued by an access control function of the QKDN controller, which manages an access control repository of registered functional components including cryptographic applications and KSAs. It is necessary to detect suspicious behaviour through authorization in the key management layer.

(2) Issues

Traditional authentication processing for cryptographic applications cannot effectively detect mass attacks, such as a denial of service attack, which uses reasonable service requests to gain excessive QKD network resources and results in the failure of access for legitimate users. Hence, ML techniques can be applied in authorizing cryptographic applications to improve the efficiency of suspicious behaviour detection in the key management layer.

(3) Role of ML in QKDN

Figure 8-3 is a schematic diagram of ML-based suspicious behaviour detection in the key management layer. First, the cryptographic application sends a key request, then ML functions observe the current key request data and analyse the behaviour. The judged result of normal or suspicious behaviour is input to the KSA. Finally, the KSA supplies keys for the authenticated normal cryptographic application. ML algorithms supporting classification, such as an artificial neural network (ANN) or recurrent neural network (RNN), can be applied.

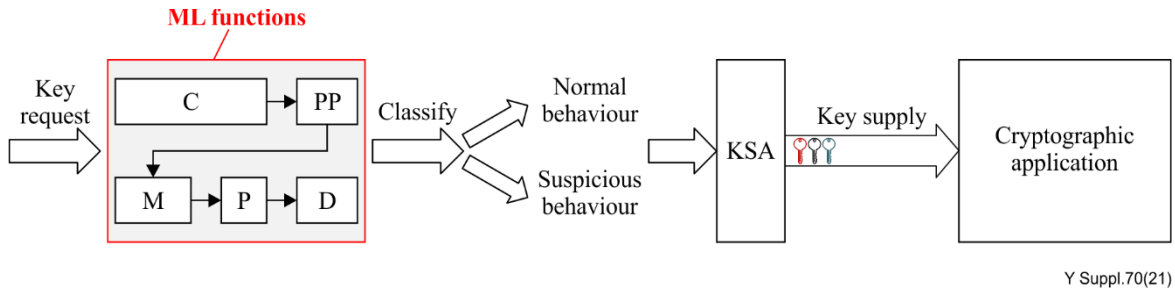

Figure 8-3 – ML-based suspicious behaviour detection in the key management layer

Use case analysis

– Analysis related to data collection follows.

  1) ML-based suspicious behaviour detection in the key management layer supports data collection in a definite long period from the cryptographic application.

  2) It collects the information related to key requests (e.g., key length, key amount, node pair names or identifiers (IDs), KSA-key ID and the key security level).

– Analysis related to data storage and processing follows.

  1) It supports storage of data used for analysis.

  2) It supports retraining the ML model after the update of datasets.

– Analysis related to application of ML output follows.

  1) The ML output is applied in authentication in the key management layer.

  2) The ML output is applied before key supply, relay or other processes that consume keys.

Benefits and impact

ML-based suspicious behaviour detection in the key management layer improves the efficiency of suspicious behaviour detection.

9 Applications of ML in the control and management layers of QKDN

9.1 Introduction

The applications of ML in the control and management layers of QKDN improvemanagement and control efficiency. Three use cases are presented, including: ML-based data collection and data pre-processing; ML-based routing; and ML-based QKDN fault prediction.

9.2 Use case CML01: ML-based data collection and data pre-processing

Use case description

(1) Background

Data collection refers to the process of gathering a system's operation information. In QKDN, the management layer collects information from other layers, as stated in [ITU-T Y.3804]. Traditional methods of data collection and pre-processing are not always adaptive, specifically in QKDNs with multi-source data. Therefore, effective categorization and aggregation of data from reference points of each layer are essential in a balanced characteristic prepared for accurate ML model training.

(2) Issues

In the QKDN management layer, data pre-processing mainly consists of data cleaning and enhancement. The collected data may include information about configuration status, network topology, inventory resources and fault records of each layer. However, the data is multi-sourced and heterogeneous. Traditional methods of data pre-processing are less efficient in associating the correlations among large-scale data, such as the network data with unexpected noise and redundancy. Hence, ML techniques can be adapted to construct a data processing model that categorizes and aggregates data into understandable, unified and easy-to-use structures.

(3) Role of ML in QKDN

The ML-based data collection and pre-processing solution collects multi-source, heterogeneous QKDN data and transforms it into understandable, unified and easy-to-use structures of data for analysis. Figure 9-1 is a schematic diagram of ML-based data collection and pre-processing. The data from the quantum, key management, service, QKDN control and QKDN management layers is input into the QKDN management layer for pre-processing. ML is mainly applied in data cleaning and data enhancement during this process.

During data cleaning, the collected data may contain the unexpected noise and redundancy. ML module is used to identify the derivative correlations between event logs and abstract the expected information. Then it removes redundant data and obtains clean data. During data enhancement, an ML module makes an effective expansion on essential data (i.e., the data with the expected information) to achieve a balance between data shortage and data redundancy. It can also avoid the uneven distribution of data features.

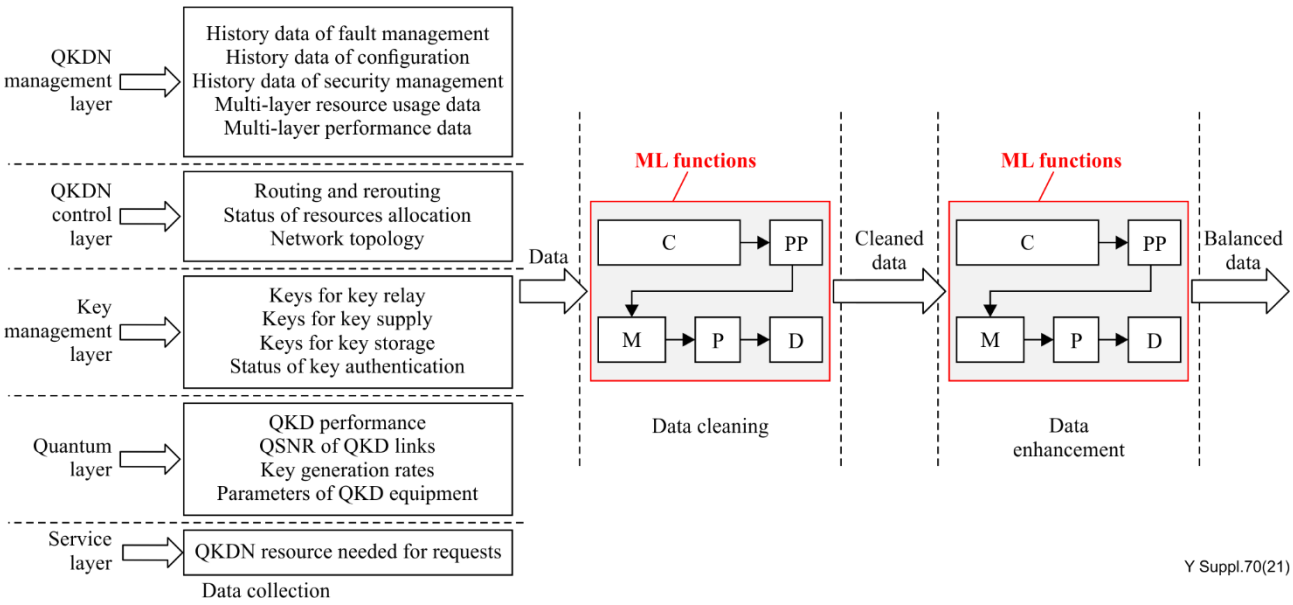

Figure 9-1 – ML-based data collection and pre-processing

Use case analysis

– Analysis related to data collection follows.

1) ML-based data collection and pre-processing collects the static data from different layers (e.g., the QKDN hardware and software data, event logs of each layer, and status of physical and virtual resources).
2) It collects the dynamic data from other layers (e.g., the performance information, configuration status, inventory and life cycle of the QKDN resources).

– Analysis related to data processing follows.

1) It supports the storage of collected data.
2) It supports the transformation of data into the expected formats.

– Analysis related to application of ML output follows.

1) The ML output is applied in the process of data cleaning to remove redundant data.
2) The ML output is applied in the process of data enhancement to achieve a balance of the data characteristics.

Benefits and impact

ML-based data collection and data pre-processing collect and pre-process multi-source, heterogeneous QKDN data in an efficient way. During data pre-processing, the collected data will be transformed into understandable, unified and easy-to-use structures and optimized in the form of balanced characteristics for the subsequent procedures.

9.3 Use case CML02: ML-based routing

Use case description

(1) Background

When a service request arrives, an appropriate route needs to be selected according to the key requirements and resource states in QKDN. The QKDN control and management layers are able to provision the key relay route. First, the two endpoint KMs inform the QKDN controller of a required number of keys from the two endpoint cryptographic applications. Then the QKDN controller analyses the status of the key management layer, especially the key consumption rate and residual

number of keys of the relevant KMs among likely candidates of key relay routes. Second, the QKDN controller finds and provisions an appropriate key relay route.

(2) Issues

Due to the dynamic and explosive nature of services, the generation and consumption of key resources are often unbalanced. When the keys on the chosen route cannot meet the key requirements of services, the success rate of services is reduced. Traditional algorithms for finding the optimal routes are computationally intensive and slow on low-power platforms. Efficient algorithms, such as ML algorithms, are needed to realize optimal routing in a reasonable amount of time.

(1) Role of ML in QKDN

Figure 9-2 is a schematic diagram of ML-based routing, which is conceived of as a classification problem. ML functions are used to classify the routing parameters. The input of ML functions can include QKD link parameters, key consumption rate and service requirements. Furthermore, the ML functions can obtain the routing configuration and real-time network re-configuration. The output of the ML functions is the optimal routing result that is used as a reference for the routing configuration.

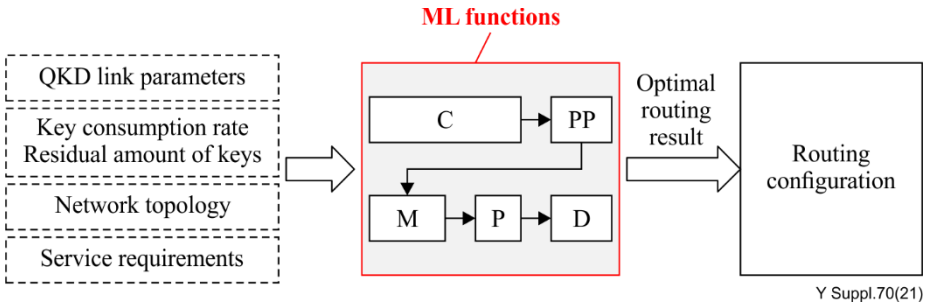

Figure 9-2 – ML-based routing

Use case analysis

– Analysis related to data collection follows.

1) ML-based routing collects information about key consumption rate and residual number of keys from the KM layer in real time.

2) It collects QKD link parameters from the QKD modules and QKDN topology information from QKDN manager in real time.

– Analysis related to data storage and processing follows.

1) It supports database update.

2) It supports pre-processing of collected data, where the data is transformed into understandable, unified and easy-to-use structures.

3) It supports retraining the ML model after the update of datasets.

– Analysis related to application of ML output follows.

1) The ML output is applied in selecting an optimal routing strategy.

2) The ML output supports ML model update under the scenarios of real-time, near real-time and non-real-time key supply for services.

Benefits and impact

The ML-based routing solution improves routing effectiveness and key resource utilization.

9.4 Use case CML03: ML-based QKDN fault prediction

Use case description

(1) Background

The QKDN control and management layers monitor QKDN performance and detect faults, as described in [ITU-T Y.3804]. If faults happen in QKDN, the communication security of services can be threatened and a large amount of data might be lost. Hence, timely prediction of faults is important according to the monitoring data and alarm information.

(2) Issues

During service provisioning, QKDN failure causes key shortage. In addition, QKDN fault recovery and service reconstruction after QKDN failure introduces time cost, which will influence the quality of services. As recommended by [ITU-T Y.3801], the QKDN manager receives the fault information provided by the QKDN controller and analyses the status information collected for fault indicators. Based on the fault information in the QKDN, ML technology is a great solution to predict QKDN faults in an effective and accurate way.

(3) Role of ML in QKDN

The ML-based QKDN fault prediction solution operates in a timely and efficient fashion. Figure 9-3 is a schematic diagram of ML-based QKDN fault prediction. ML functions collect historical alarm information in QKDN and a large amount of data in the current QKDN. Through fault classification induction and feature analysis, the ML functions are aware of the association rules between the alarm information and various faults in QKDN. The output of ML functions is the fault prediction results, including their type, number, location and time. A convolutional neural network has few parameters, but powerful feature extraction and representation ability. A fully connected network having a simple structure can comprehensively extract potential valuable information from data. The two ML models are suggested for use to build a mathematical model of fault prediction.

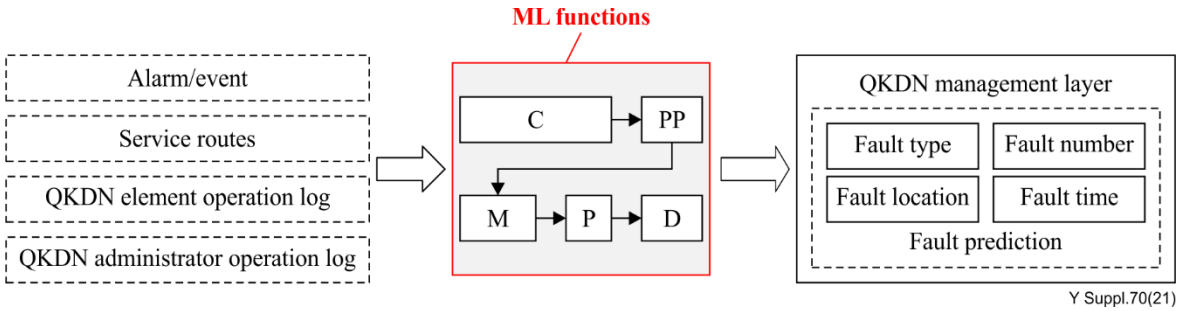

Figure 9-3 – ML-based QKDN fault prediction

Use case analysis

– Analysis related to data collection follows.

  1) ML-based QKDN fault prediction collects historical alarm information from the QKDN control, key management and quantum layers.

  2) It collects the operation and alarm data in the current QKDN (e.g., alarm, service routes, QKDN element operation log and QKDN administrator operation log).

– Analysis related to data storage and processing follows.

  1) It supports the storage of data used for analytics.

  2) It supports data cleaning and data enhancement for large amounts of collected data.

  3) It supports retraining the ML model after the update of datasets.

– Analysis related to application of ML output follows.

1) The ML output is applied in fault prediction.
2) The ML output is applied in supporting survivability protection in advance before QKDN faults happen.

Benefits and impact

The ML-based QKDN fault prediction solution operates in a timely and efficient fashion to avoid the loss and risk of QKDN faults.

Bibliography

[ITU-T Y-Suppl.55] ITU-T Y-series Recommendations – Supplement 55 (2019), *ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: use cases*.

[b-ETSI GR ENI 004] ETSI Group Report ENI 004 V2.1.1 (2019), *Experiential networked intelligence (ENI); Terminology for main concepts in ENI*.

[b-ETSI GR QKD 007] ETSI Group Report QKD 007 V1.1.1 (2018), *Quantum key distribution (QKD); Vocabulary*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021