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

Quantum Key Distribution Networks - Applications of Machine Learning

Summary

For quantum key distribution networks (QKDN), the supplement 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, issue, role of ML in QKDN, use case analysis and, benefits and impact.

Keywords

Applications; Machine learning (ML); Quantum key distribution (QKD); QKD networks.

Table of Contents

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

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, the scope of this draft supplement will include:

- 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.3800] Recommendation ITU-T Y.3800 (2019)/Cor.1 (2020), *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*.

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

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

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 – 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 [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 (QKD) [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 (QKD 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

This chapter defines all the terms used in this supplement.

None.

4. Abbreviations and acronyms

This supplement uses the following abbreviations and acronyms:

AES Advanced Encryption Standard

ANN Artificial Neural Network

KM Key Manager

KMA Key Management Agent

KSA Key Supply Agent

LSTM Long Short-Term Memory

ML Machine Learning

QBER	Quantum Bit-Error Ratio
QKD	Quantum Key Distribution
QKDN	Quantum Key Distribution Network
OSNR	Optical Signal-to-Noise Ratio
RNN	Recurrent Neural Network
RUL	Remaining Use Life
SPD	Single Photon Detector

5. Conventions

None.

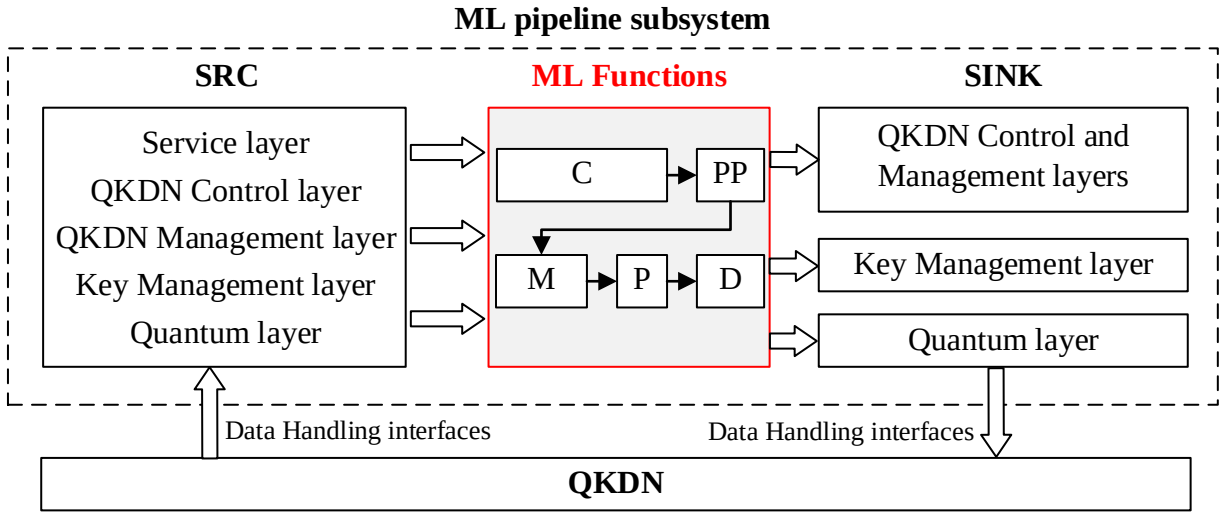
6. Overview

Quantum key distribution network (QKDN) is a technology that extends the reachability and availability of quantum key distribution (QKD), which is stated in [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 will increase as the scale of the network increases.

Machine learning (ML) mechanisms are able to teach a computer to learn knowledge using data without being explicitly programmed. ML can be applied to 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 traffic prediction and fault prediction. There is increasing interest and necessity in applying ML to improve the performance of QKDNs. Due to the advantages of ML, ML can be applied in QKDNs so as to improve the QKD performance and the control and management efficiency of QKDN.

ML pipeline subsystem in QKDN is shown in Fig.6.1. The ML functions support a set of functional elements in a 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 source of data (SRC) through data handling interfaces. The SRC can be in different layers of QKDNs. The target of the ML output (SINK) can be elements in quantum layer, key management layer and QKDN control and management layers. More details related to ML pipeline subsystems can be found in [ITU-T Y.3172].

In the following clauses, based on the academic and industrial advances, the supplement presents several applications of ML in the quantum layer, in the key management layer and in the management and control layers of QKDN including the use case background, issue, role of ML in QKDN, use case analysis and, benefits and impact.



C: collector; PP: preprocessor; M: model; P: policy; D: Distributor; SRC: source of data; SINK: target of ML output

Fig. 6.1. ML pipeline subsystem in QKDN

7. Applications of ML in the quantum layer of QKDN

7.1. Introduction

The applications of ML in the quantum layer of QKDN represent applying the ML to improve the performance of the quantum layer. 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 implementing and commercializing QKDNs. The main challenge is that the noise will fall into the quantum channel, thereby reducing the quality of quantum channel and causing low key rate, especially when the quantum encoded photons coexist with high-intensity classical signals. Recently, ML-based techniques have been applied to optical communication to predict optical signal-to-noise ratio (OSNR).

(2) Issue

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

(3) Role of ML in QKDN

During the QKD process, the noise falling in the quantum channel will cause the deterioration of the quantum channel quality and the generated key rate. Fig 7.1 shows the diagram of ML-based quantum channel performance prediction. Firstly, the ML functions collect the quantum channel related data

and the corresponding quantum channel performance through quantum channel measurement for ML model training and testing. Then, with the trained ML model, the quantum channel performance can be predicted based on the current input quantum channel related data. Lastly, according to the predicted channel performance, feedback adjustment can be done in advance to improve the channel environment and reduce unnecessary loss caused by key rate decreases.

Supervised ML method can be beneficial to estimate the quantum channel performance (noise, quantum bit-error ratio (QBER), etc.) when various quantum channels, allocated spectrum, launch power and channel spacing exist.

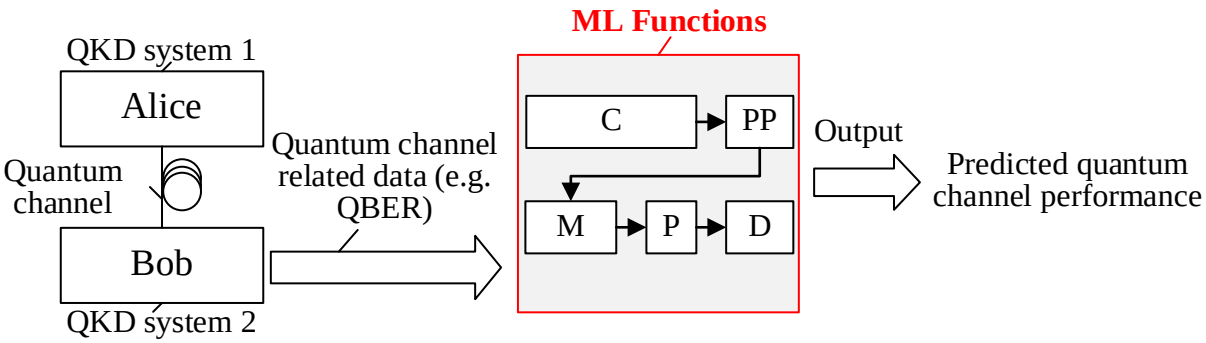


Fig 7.1. ML-based quantum channel performance prediction

Use case analysis

- Analysis related to data collection

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

- Analysis related to data storage and processing

  1) It 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

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

Benefits and impact

ML-based quantum channel performance prediction method will predict the quantum channel performance under different channel noise environments. Measures can be taken in advance based on the predictions to improve the channel environment and make 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 very important to keep QKD system under the best performance and necessary to optimize the parameters of the QKD system. QKD system parameters include the intensities of signal and decoy state and the probabilities to choose different intensities and bases, as well as the probability that QKD receiver Bob measures the incoming pulse with Z basis, etc.

(2) Issue

Practically, the insufficient computing power of a QKD system will cause either waiting for an optimization off-line (and suffer from delay) or using suboptimal or even unoptimized parameters in real time, which will reduce the efficiency of the basis-sift factor. At the same time, when the gain and QBER change with the environment, the parameters of the QKD system 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 to obtain the optimal parameters of QKD systems by learning from a large number of training data, which will efficiently realize the optimization of a QKD system.

(3) Role of ML in QKDN

The ML-based QKD system parameter optimization solution is to pre-execute optimization algorithm before key generation. Fig. 7.2 shows the diagram of ML-based QKD system parameter optimization. Firstly, the input data is sampled to pick a random combination of physical parameters which cannot be controlled by the users, and use a local search algorithm to calculate their corresponding optimization parameter values which can be adjusted by the users. The obtained physical parameter values are inputted into ML model trainer with the selected ML model. After the training, N sets of prediction parameter values are output. By comparing the key rate obtained by the classical algorithm with the key rate based on the predicted parameter values, the comparison result is fed back to the ML model trainer. Secondly, when the QKD system needs parameter optimization, the real-time data is inputted and the optimal parameter values are outputted after applying the ML functions. Finally, the configuration parameters are input into the QKD system to complete parameter optimization.

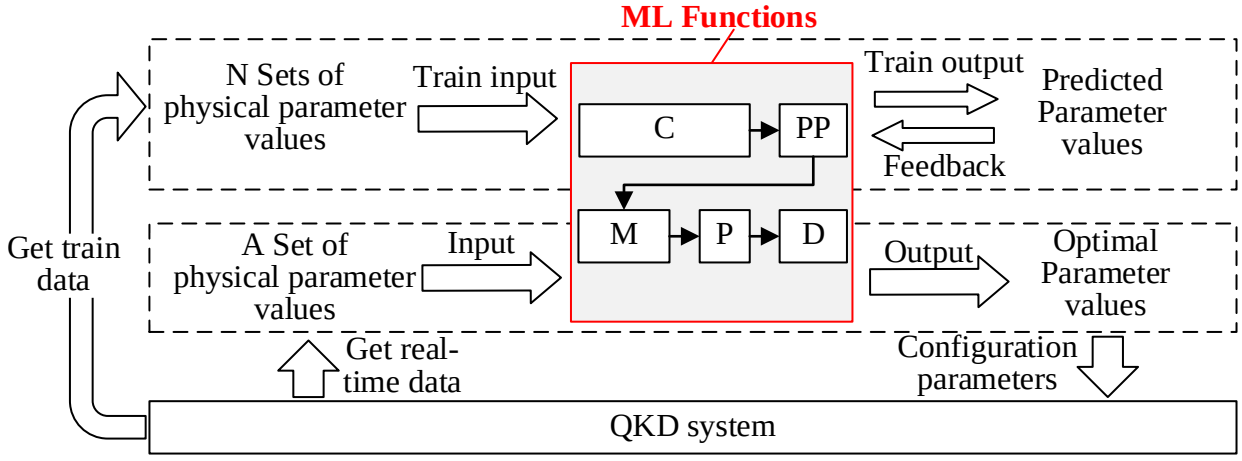


Fig. 7.2 ML-based QKD system parameter optimization

Use case analysis

- Analysis related to data collection

  1) It 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, the number of signals) in the quantum layer.

- Analysis related to data storage and processing

1) It supports simple scaling and normalization of input data
2) It supports real-time prediction which aims to predict optimal parameters.
3) It supports retraining the ML model after the update of datasets.

- Analysis related to application of ML output

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

Benefits and impact

The ML-based QKD system parameter optimization solution will optimize the QKD system 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 remaining use life (RUL) prediction of components in a QKD system

Use case description

(1) Background

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

(2) Issue

Considering the influence of external environments and internal factors, we don't know when the components will stop working. The components of a QKD system include pulsed light source, decoy state modulation module, random number generator, SPD and so on. It is very important to make an accurate prediction of the RUL of components in QKD systems. Due to the advantages of ML, ML technique has the ability to predict RUL of components under large number of RUL-related data.

(3) Role of ML in QKDN

Fig. 7.3. shows the diagram of ML-based RUL prediction of components in a QKD system. Firstly, component parameters in a QKD system such as temperature, power and electricity are collected, classified and pre-processed. Then, the pre-processed data is input into the ML functions. After the ML model training, the ML output is achieved. Finally, the predicted RUL is output. Considering the long life of laser and the prediction target, long short-term memory (LSTM) is suggested to implement the RUL prediction of laser.

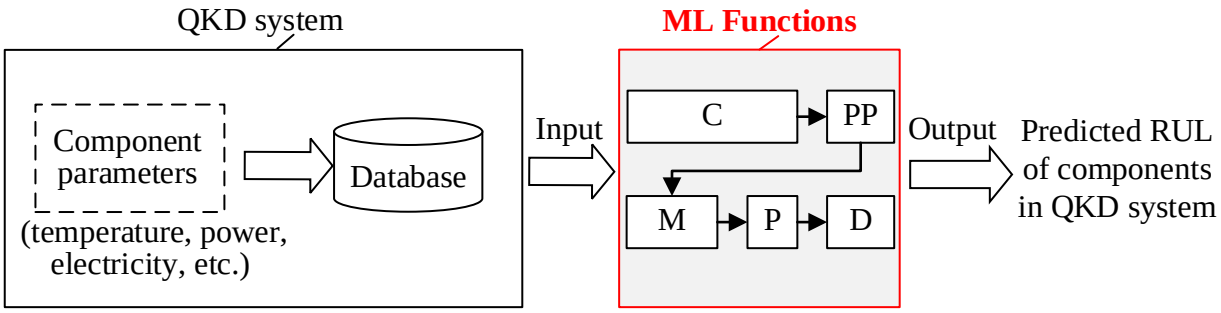


Fig. 7.3 ML-based RUL prediction of components in a QKD system

Use case analysis

- Analysis related to data collection

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

- Analysis related to data storage and processing

1) It supports storage of the measured values used for analysis.
2) It supports the data processing such as normalization and segmentation.

- Analysis related to application of ML output

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

Benefits and impact

The ML-based RUL prediction of components in a QKD system solution will accurately estimate the RUL of components, which helps to ensure the stable QKD system operations.

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

8.1. Introduction

The applications of ML in the key management layer of QKDN represent applying the ML in the key management layer and improving the key management efficiency and stability. Three use cases are presented including ML-based key formatting, ML-based key storage management, and ML-based suspicious behavior detection in the key management layer.

8.2. Use case KM01: ML-based key formatting

Use case description

(1) Background

When supplying keys or relaying keys, keys will need to be combined or split where the lengths of keys are not appropriate, which is stated in [ITU-T Y.3803]. There are different key formats for different security requirements of services and different encryption algorithms (e.g. OTP, AES-512, AES-256, AES-128). To maintain interconnectivity and expandability in the QKDN, proper key format needs to be solved for key data with added metadata containing various types of information.

(2) Issue

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

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

(3) Role of ML in QKDN

The ML-based key formatting solution is to guide the key formatting before storing keys with the awareness of the service characteristics. Fig. 8.1 shows the 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 keys and then store keys in the key store for the future key supply. Since the number of the stored keys with definite formats is based on the service characteristics, the times of key re-formatting will reduce while supplying keys for services. As for the ML models, the prediction models such as deep learning algorithms and Elman neural network can be applied.

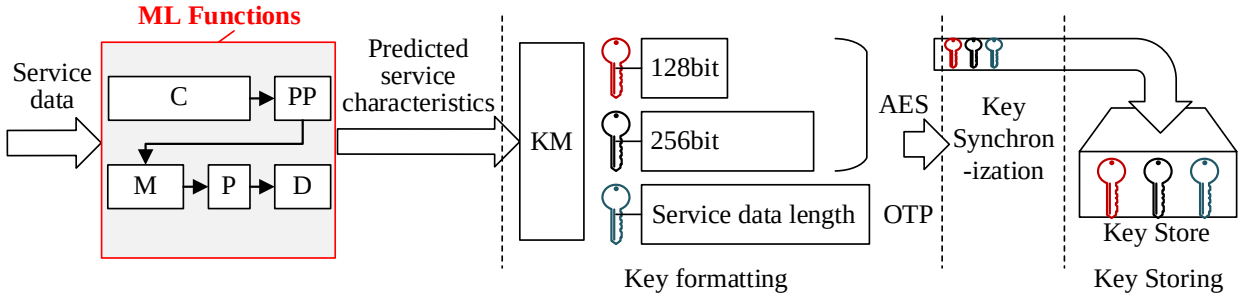


Fig. 8.1 ML-based key formatting

Use case analysis

- Analysis related to data collection

  1) It 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 ML model in real time.
  2) The collected service data includes the service characteristics (e.g. the service arriving time, the service duration time, the service security levels and the size of service data which 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

  1) It supports storage of data used for analysis.
  2) It supports real-time prediction which aims to predict 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

1) The ML output is applied in the process of key formatting before storing keys.
2) The KM is able to operate key formatting according to the ML output.

Benefits and impact

The ML-based key formatting solution will guide the key formatting with the awareness of service characteristics before storing keys, which will reduce the time cost and the risk of key non-synchronization failure during the 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, which is stated in [ITU-T Y.3800].

(2) Issue

The key requirement of services changes dynamically in the actual situation. On the one hand, services may not be supplied with keys successfully because of many factors including insufficient key resources in key storage, keys with long storage time and so on. On the other hand, 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, the existing traditional solutions are difficult to accurately perceive the actual needs of services and establish the evaluating 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 is to reasonably evaluate and predict health state of key storage. The diagram of ML-based key storage management is shown in Fig. 8.2. ML functions collect a lot of training data, uses ML algorithms to build ML model, and constantly adjusts it through the result comparison, so as to accurately evaluate and predict health states of key storage. KM manages keys in key storage based on the output value of ML functions, so as to ensure reasonable scheduling, efficient utilization of key resources and avoid problems such as long key storage time or excessive jitter. The KMA manages the key life cycle, which is to archive keys or to destroy the keys which have been stored for a long time. Finally, keys are supplied to cryptographic application on demand if the key storage is in a health state.

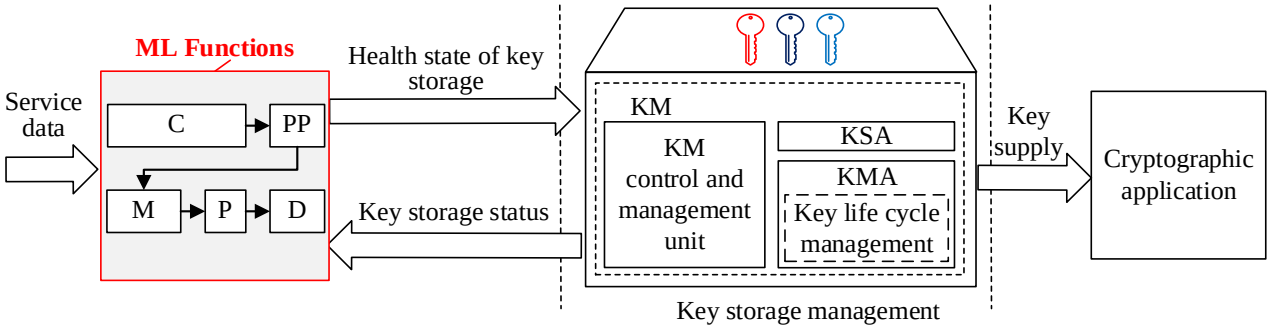


Fig. 8.2 ML-based key storage management

Use case analysis

- Analysis related to data collection

  1) It 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

  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

  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

ML-based key storage management solution will evaluate and predict health state of key storage and help to realize the efficient utilization of key resources.

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

Use case description

(1) Background

Suspicious behavior detection exists in the quantum layer and key management layer, and this use case focuses on the second scenario. The 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’s necessary to detect suspicious behavior detection through authorization in the key management layer.

(2) Issue

However, the traditional authentication processing for cryptographic applications can't effectively detect mass attacks, such as DoS attack which uses reasonable service requests to gain excessive QKD network resources and results in the failure of accessing by legitimate users. Hence, ML techniques can be applied into authorizing cryptographic applications to improve the efficiency of suspicious behavior detection in the key management layer.

(3) Role of ML in QKDN

The diagram of ML-based suspicious behavior detection in the key management layer is shown in Fig. 8.3. First of all, the cryptographic application sends a key request, and then ML functions observe the current key request data and analyze the behavior. The judged result of normal behavior or suspicious behavior is input to KSA. Finally, the KSA supplies keys for the authenticated normal cryptographic application. ML algorithms supporting classification such as artificial neural network (ANN), recurrent neural network (RNN) can be applied.

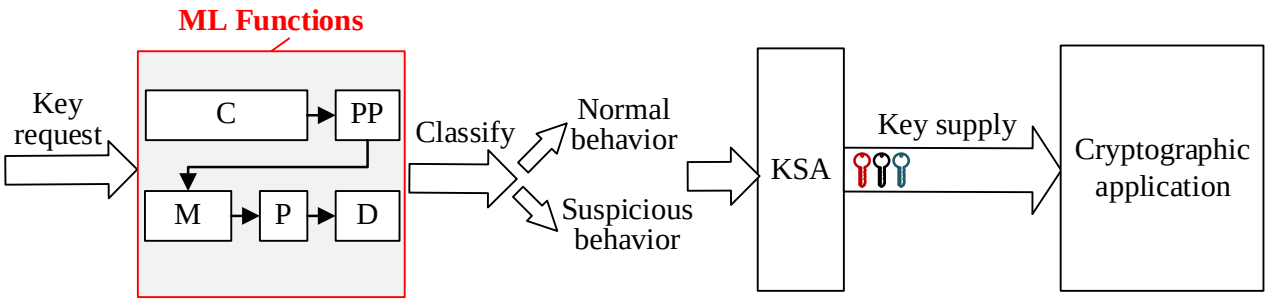


Fig. 8.3 ML-based suspicious behavior detection in the key management layer

Use case analysis

- Analysis related to data collection

  1) It supports the 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 IDs, KSA-key ID, and the security level of key).

- Analysis related to data storage and processing

  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

  1) The ML output is applied in the authentication in the key management layer.
  2) The ML output is applied before the key supply, key relay or other processes to consume keys.

Benefits and impact

ML-based suspicious behavior detection in the key management layer will improve the efficiency of suspicious behavior 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 represent applying the ML in the control and management layers and improving the QKDN management 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. As for the data collection in QKDN, the QKDN management layer collects information from other layers, which is 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, it will be an essential progress to effectively categorize and aggregate the data from reference points of each layer in a balanced characteristic prepared for the accurate ML model training.

(2) Issue

In the QKDN management layer, data pre-processing is mainly consisted of data cleaning, and data enhancement. The collected data may include the information of configuration status, network topology, inventory resources and fault records of each layer and so on. However, the data is multi-sourced and heterogeneous. Traditional methods of data pre-processing are less efficient in associating the correlations among the 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 is to collect multi-source, heterogeneous QKDN data and transformed it into understandable, unified and easy-to-use structures of data for analysis. Fig. 9.1 shows the diagram of ML-based data collection and pre-processing. The data from quantum layer, key management layer, service layer, QKDN control layer and QKDN management layer is inputted 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 gets the clean data. During data enhancement, 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.

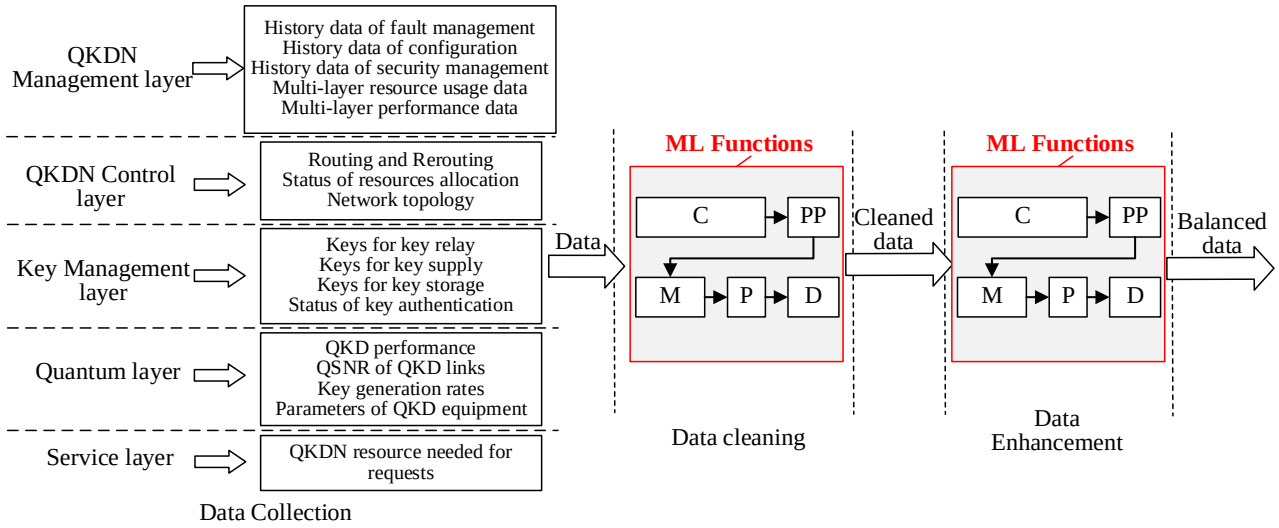


Fig. 9.1 ML-based data collection and pre-processing

<u>Use case analysis</u>

- Analysis related to data collection

  1) It 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

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

- Analysis related to application of ML output

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

Benefits and impact

The ML-based data collection and data pre-processing will collect and pre-process multi-source, heterogeneous QKDN data in an efficient way. During the data preprocessing, 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. Firstly, the two end-point 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 along likely candidates of key relay routes. Finally, the QKDN controller finds and provisions an appropriate key relay route.

(2) Issue

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 will be reduced. Traditional optimization 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.

(3) Role of ML in QKDN

Diagram of ML-based routing is shown in Fig 9.2. The ML-based routing is thought as a classification problem. The ML functions are used to classify the routing parameters. The input of ML functions could include QKD link parameters, key consumption rate, service requirements and so on. Furthermore, the ML functions can obtain the routing configuration and real-time network re-configuration. The output of ML functions is the optimal routing result which will be used as a reference for routing configuration.

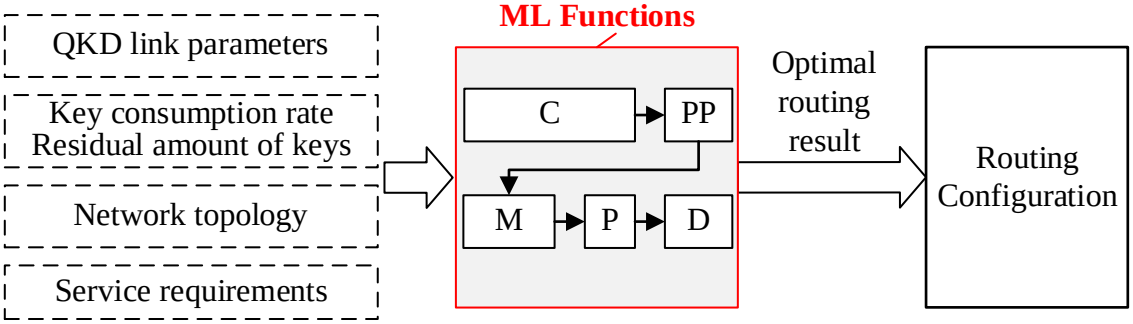


Fig. 9.2 ML-based routing

Use case analysis

- Analysis related to data collection

  1) It 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

  1) It supports to update the database.
  2) It supports to pre-process the 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

  1) The ML output is applied in selecting an optimal routing strategy.
  2) The ML output supports to update the ML model under the scenarios of real-time, near real-time and non-real-time key supply for services.

Benefits and impact

ML-based routing solution will improve the routing effectiveness and the key resources utilization.

9.4. Use case CML03: ML-based QKDN fault prediction

Use case description

(1) Background

The QKDN control and management layers monitor QKDN performances and detect faults, which is described in [ITU-T Y.3804]. If faults happen in QKDN, the communication security of services can be threatened and a large amount of loss might be caused. Hence, it is very important to predict faults in time according to the monitoring data and alarm information.

(2) Issue

During service provisioning, QKDN failure will cause key shortage. And, QKDN fault recovery and service reconstruction after QKDN failure will introduce 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 QKDN controller and analyses the status information collected for fault indicators. Based on the fault information in QKDN, ML technology is a great solution to realize QKDN fault prediction in an effective and accurate way.

(3) Role of ML in QKDN

The ML-based QKDN fault prediction solution is to predict faults in QKDN timely and efficiently. The diagram of ML-based QKDN fault prediction is shown in Fig. 9.3. ML functions collect history alarm information in QKDN and a large amount of data in the current QKDN. Through the 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 fault type, fault number, fault location, fault time and so on. Convolutional neural network has few parameters but powerful feature extraction and representation

ability. Full-connected network having simple structure can fully extract potential valuable information from data. The two ML models are suggested to be used to build the mathematical model of fault prediction.

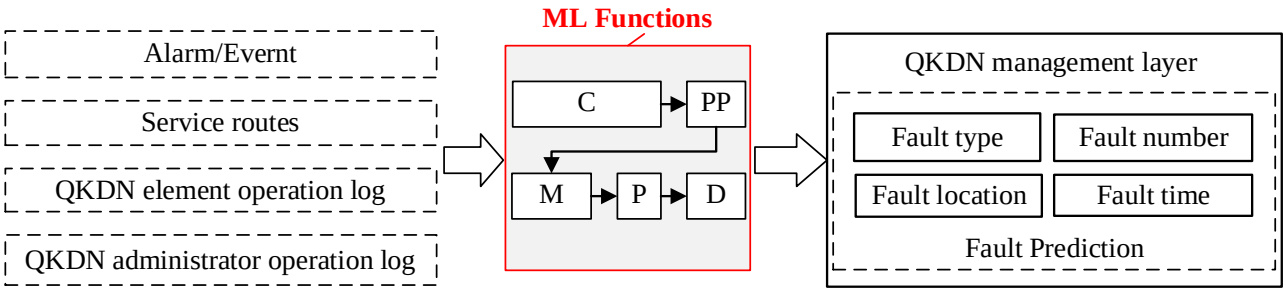


Fig. 9.3 ML-based QKDN fault prediction

Use case analysis

- Analysis related to data collection

  1) It collects the history alarm information from the QKDN control layer, key management layer and quantum layer.
  2) It collects the operation and alarm data in the current QKDN (e.g. alarm, service routes, QKDM element operation log and QKDN administrator operation log).

- Analysis related to data storage and processing

  1) It supports the storage of data used for analytics.
  2) It supports data cleaning and data enhancement for the large amount of the collected data.
  3) It supports retraining the ML model after the update of datasets.

- Analysis related to application of ML output

  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 will realize fault prediction timely and efficiently to avoid the loss and the risk of QKDN faults.

Bibliography

[b-ETSI GR ENI 004]

ETSI GR ENI 004 V1.1.1 (2018), Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI.

[ITU-T Y Suppl. 55]

Supplement to ITU-T Y.3170-series (2019), Machine learning in future networks including IMT-2020: use cases.

[ETSI GS QKD 002]

ETSI GS QKD 002 (2010), Quantum Key Distribution; Use Cases.