Titre: Title:	Réalisation d'un GAN quantique adapté à la détection d'anomalies pour les séries temporelles issues de données réseau
Auteur: Author:	Benjamin Kalfon
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Kalfon, B. (2024). Réalisation d'un GAN quantique adapté à la détection d'anomalies pour les séries temporelles issues de données réseau [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/58023/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/58023/
Directeurs de recherche: Advisors:	Soumaya Cherkaoui
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Réalisation d'un GAN quantique adapté à la détection d'anomalies pour les séries temporelles issues de données réseau

BENJAMIN KALFON
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Avril 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Réalisation d'un GAN quantique adapté à la détection d'anomalies pour les séries temporelles issues de données réseau

présenté par **Benjamin KALFON**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Alejandro QUINTERO, président
Soumaya CHERKAOUI, membre et directrice de recherche
Heng LI, membre

DÉDICACE

À noha et titi.

REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont participé au projet de recherche, que ce soit ma directrice de recherche Soumaya Cherkaoui ou tous les membres m'ayant aidé. Remerciements spéciaux à ceux qui m'ont donnés des retours pour la rédaction de l'article : Jean Frédéric Laprade, Ola Ahmad et Shengrui Wang. Je remercie également mes amis et mes colocataires qui m'ont fait passer deux ans agréables à Montréal.

RÉSUMÉ

Les séries temporelles sont un type de données omniprésent dans le monde scientifique. De par leur nature temporelle, elles sont utilisées pour enregistrer l'évolution de toutes sortes de phénomènes au cours du temps, que ce soit en météorologie, en finance, en réseautique. Dès qu'on veut surveiller l'activité d'un système, ce type de donnée est utile. De part leur omniprésence dans des domaines cruciaux, il devient intéressant de savoir détecter des anomalies pour ce type de données. Typiquement, un dysfonctionnement du système dont on extrait une série temporelle va se manifester dans ladite série. Détecter ce changement comme anormal permet d'être alerté du dysfonctionnement. Par exemple, une attaque par déni de service distribué ouDistributed Denial of Service (DDoS) sur un réseau va se traduire par un pic bref dans la série temporelle du nombre de paquets reçu par le réseau. Dans ce contexte particulier de la sécurité réseau, il y a d'énormes enjeux à détecter les anomalies puisque toutes les infrastructures importantes de la société sont basées sur des réseaux informatiques. En conséquence, le nombre de cyber-attaques sur des réseaux ne fait qu'augmenter années après années.

Il existe dans la littérature de nombreuses approches génériques de détection d'anomalies, applicables en particulier à des séries temporelles. Nombre d'entre elles se basent sur de l'apprentissage machine, plus particulièrement sur de l'apprentissage profond et donc les réseaux de neurones. Typiquement, on peut utiliser des approches de classification : en entraînant un agent sur les données normales et différentes classes de données anormales (types de dysfonctionnements, types d'attaques), l'agent devient capable de reconnaître les signatures de différentes anomalies dans les données. Ces approches d'apprentissage supervisé ont le désavantage de limiter le système de détection aux anomalies déjà rencontrées et répertoriées en suffisamment grand nombre pour pouvoir avoir assez de données pour l'entraînement souvent coûteux de réseaux de neurones. Pour éviter cet écueil, une autre approche développée dans la littérature est d'utiliser un modèle génératif, généralement un réseau antagoniste génératif ou Generative Adversarial Network (GAN). Le principe est d'entraîner un modèle capable de générer des données en accord avec la distribution des données d'entraînement. Le modèle est donc entraîné à générer les données normales, puis en comparant la prédiction du modèle avec des données réelles, on peut estimer un score d'anomalie.

Un autre aspect de la détection d'anomalies est son caractère ultra-compétitif. Au vu des enjeux présentés plus haut, il y a d'énormes incitations à mettre au point des mécanismes d'attaques sur les infrastructures réseau toujours plus efficaces et donc d'énormes raisons

de développer des systèmes de détection d'anomalies à la hauteur. Il est donc nécessaire d'explorer toutes les voies technologiques nouvelles ayant un potentiel pour cette application. L'une de ces voies émergentes est le calcul quantique.

Depuis quelques années, des progrès significatifs ont été faits dans ce domaine en termes de matériel, nous rapprochant de plus en plus de la mise en application d'algorithmes exponentiellement plus efficaces pour traiter certaines taches. Dans cet effort d'exploration de la technologie, des travaux ont laissé entrevoir la possibilité de bénéfices apportés par l'informatique quantique dans le cadre de l'apprentissage machine. Plusieurs approches utilisent un circuit variationnel quantique ou Quantum Variational Circuit (QVC), qui est l'équivalent quantique des agents paramétriques classiques comme les réseaux de neurones, sont apparues dans la littérature. Une de ces approches est celle basée sur un réseau antagoniste génératif quantique ou Quantum Generative Adversarial Network (QGAN), ce domaine s'est développé lors des dernières années depuis les deux articles fondateurs en 2018. Cependant, ce type de modèle n'a pas encore été appliqué à la détection d'anomalies pour les séries temporelles en particulier.

C'est donc cette piste que ce travail de maîtrise cherche à explorer, en ayant pour but d'implémenter un QGAN de l'état de l'art pour la détection d'anomalies pour des séries temporelles issues de données réseau. De nombreux défis se manifestent pour cette application particulière, tout particulièrement des défis dus à la dimension des données : une série temporelle est typiquement constitué d'un nombre de points trop élevé pour les limitations matérielles des ordinateurs quantiques actuels et dans un futur proche.

L'approche présentée dans ce mémoire contourne ce défi de la dimensionnalité des données par deux moyens, l'utilisation d'un réseau antagoniste génératif conditionnel ou conditional Generative Adversarial Network (cGAN) et un nouveau design de circuit quantique pour l'encodage de données hautes dimension dans un état quantique : l'injection successive de données ou Successive Data Injection (SuDaI). Le QGAN est ensuite utilisé au sein d'un système de détection d'anomalies dont on fait la démonstration sur le jeu de données AWSCloudWatch du Numenta Anomaly Benchmark (NAB) qui est constitué de séries temporelles issues d'enregistrement d'un indicateur clé de performance ou key performance indicator (kpi) sur des serveurs comme par exemple la consommation d'un processeur. On utilise de l'apprentissage en continu sur les séries pour détecter des points marqueurs d'anomalie.

ABSTRACT

Time series is a type of data that appears everywhere in the scientific and industrial world. Because of their temporal nature they are used to monitor the evolution of many different phenomenons over time in many different fields such as meteorology, finance or networking. Whenever one wants to monitor the activity of a system, this type of data proves to be useful. Because of this presence in such crucial domains, it is very interesting to be able to detect anomalies in this type of data. Typically a malfunctioning in a system monitored by a time series will be translated by an anomaly in the said series. Detecting this change as abnormal in the time series warns about the malfunctioning. For example, a DDoS on a network will be translated into a spike in the time series of the number of packets received by the network. In this particular context of network security, the stakes are high when it comes to detecting anomalies, since most important infrastructures rely on computer networks. As a consequence, the number of cyber-attacks on networks only raises year after year.

In the literature, there are numerous approaches for general anomaly detection that consequently can be applied to time series. Many of them are based on machine learning, more precisely on deep learning and on neural networks. Typically, one can use classification methods: by training an agent on normal data and different classes of anomalous data, the agent is then able to recognize the signatures of different anomalies in the data. Such supervised learning methods have the drawback of limiting the detection system to anomalies that are already well documented in large enough numbers to be able to perform the often costly training of the neural networks. To avoid this, another approach developed in the literature is to use a generative model, generally a GAN. The idea is to train a model to generate data following the distribution of the training data. The model is thus trained to generate normal data then by comparing this forecast with real data one can estimate an anomaly score.

Another aspect of anomaly detection is its highly competitive aspect. In the light of the stakes presented earlier, there are huge incentives in the direction of developing more and more efficient attack mechanisms on network infrastructures. Consequently, there also are huge incentives for the development of anomaly detection systems with more and more capability. It is thus necessary to explore any technology that could potentially lead to improvements in this field. One of these promising and emerging fields is quantum computing.

In recent years, significant progress has been made in this domain in terms of hardware, taking us ever closer to the implementation of exponentially more efficient algorithms than their classical counterparts for certain types of applications. In this effort of exploration of

quantum computing, works have hinted at the possibility of advantages that it could yield in the frame of machine learning. Different approaches using a quantum variational circuits or QVC, which is the quantum equivalent of classical parametric agents such as neural networks have appeared in the literature. More specifically, the field of QGANs has developed in the last few years since the two foundational articles in 2018. However, this type of model has not yet been applied to time series anomaly detection.

It is this path that this master's thesis seeks to explore, with the aim of implementing a state-of-the-art QGAN for anomaly detection for time series monitoring network data. Many challenges arise from this particular application,most notably challenges coming from the dimension of the data: a time series typically consists of a number of points too large for the hardware limitations of current and near-future quantum computer as well as quantum simulators. The approach presented in this thesis circumvents this data dimensionality challenge in two ways, the use of a cGAN and a new quantum circuit design for encoding high-dimensional data into a quantum state: successive data injection or SuDaI. The QGAN is then used inside an anomaly detection that we demonstrate on the dataset AWSCloudwatch from the Numenta Anomaly Benchmark or NAB that consists of time series monitoring key performance indicators or kpis of servers such as CPU usage. We use continual learning on the time series to detect anomalous points.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

cGAN	conditional Generative Adversarial Network
DDoS	Distributed Denial of Service
GAN	Generative Adversarial Network
kpi	key performance indicator
NAB	Numenta Anomaly Benchmark
QGAN	Quantum Generative Adversarial Network
QVC	Quantum Variational Circuit
SuDaI	Successive Data Injection

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Contexte et problématique

La détection d'anomalies pour les séries temporelles est un enjeu crucial de nombreux domaines scientifiques et industriels. Dès lors que l'on veut étudier l'évolution d'un système, on enregistre des grandeurs en fonction du temps, ce qui produit une série temporelle. Des systèmes critiques comme le climat, les plaques tectoniques, les institutions financières, les réseaux importants sont aujourd'hui surveillés par des séries temporelles et il y a d'énormes enjeux à réussir à prévenir des dysfonctionnements ou attaques sur ces systèmes, car l'impact d'anomalies peut potentiellement être dévastateur [1–5]. En conséquence, l'analyse des séries temporelles dans le but de détecter des anomalies est un champ étudié depuis plusieurs décennies [6, 7].
L'importance de cette tâche tend à encore augmenter dans le domaine du réseau et de la cybersécurité au vu de l'explosion des infrastructures numériques des dernières années tant en nombre qu'en taille. En conséquence, on observe donc l'explosion de données, en particulier temporelles, liées a ces infrastructures. En comparaison avec d'autres domaines, la sécurité des systèmes informatiques possède la caractéristique supplémentaire d'être hautement compétitive. En effet, si les techniques de détection d'anomalies s'améliorent, les méthodes d'attaque continuent toujours d'évoluer. Dans ce contexte adversariel, il est intéressant d'explorer toutes les pistes potentielles qui pourraient se révéler utiles.

Une piste intéressante est la technologie émergente et prometteuse qu'est l'informatique quantique. Ces dernières années, les progrès matériels dans la réalisation d'ordinateurs quantiques ont permis des avancées démontrant un avantage des technologies quantiques par rapport aux ordinateurs classiques [8]. C'est dans ce contexte que s'inscrit ce travail de maîtrise. L'idée initiale est d'adapter des approches de l'état de l'art classique pour la détection d'anomalies appliquée aux séries temporelles au cadre de l'informatique quantique.
Une grande partie de l'état de l'art dans la détection d'anomalies en général et de la détection d'anomalies pour les séries temporelles en particulier utilise des méthodes basées sur l'apprentissage machine et l'apprentissage profond [9, 10]. Parmi elles se trouvent les méthodes basées sur les GANs [11, 12]. Ces méthodes ont l'avantage de ne pas nécessiter d'apprentissage supervisé : on n'a pas besoin d'exemples d'anomalies pour développer le système de détection. C'est sur ce type de méthode que le choix de recherche s'est porté puisqu'il existe déjà des propositions de QGANs [13–17].

Cependant, la plupart de ces propositions sont démontrées expérimentalement sur un nombre de qubits réduit dû aux paradigmes généraux employés pour la représentation des données dans l'ordinateur quantique. Cette limitation est couplée à une autre limitation : la limitation matérielle des ordinateurs quantiques qui restreint le nombre de bits quantiques ou qubits utilisables. On peut, pour essayer de contourner cette difficulté, d'implémenter les lois mathématiques qui régissent un ordinateur quantique dans un ordinateur classique et ainsi créer un simulateur classique. Si ces derniers permettent de réaliser les mêmes fonctions que les véritables ordinateurs quantiques, ils ne sont pas non plus une solution puisque le coût de simulation est exponentiel en fonction du nombre de qubits. Or, les données comme les séries temporelles sont typiquement composées d'un très grand nombre de points. Il est donc extrêmement difficile d'appliquer les méthodes de la littérature à des séries temporelles pour faire de la détection d'anomalies. C'est ce point qu'il s'agira de résoudre au long de ce travail de maîtrise.

1.2 Objectifs de recherche

Ce projet de recherche est issu d'un partenariat entre polytechnique Montréal, l'université de Sherbrooke, THALES DIGITAL SOLUTIONS Inc. et ZETANE SYSTEMS Inc.

L'objectif principal est l'exploration des méthodes basées sur les QGANs pour la détection d'anomalies pour les séries temporelles. Il se décompose en 2 axes principaux :

- Explorer et proposer des solutions au défi de la dimension des données des données dans le contexte des contraintes de l'informatique quantique.
- Incorporer la solution en une proposition complète de système de détection d'anomalies temporelle utilisant un QGAN issu de la littérature récente.

1.3 Plan du mémoire

La suite de ce mémoire se décompose de la manière suivante : le chapitre 2 est une revue de littérature présentant les travaux de détection d'anomalies utilisant des GANs, les travaux sur l'apprentissage machine quantique et les QVCs ainsi que les propositions de QGANs. Le chapitre 3 fournit les connaissances utiles pour comprendre la proposition et la démarche associée. Le Chapitre 4 est l'article présentant la proposition et les résultats. Le chapitre 5 contient une discussion sur ce travail de recherche qui explique les choix faits en fonction des

difficultés rencontrées. Enfin, le chapitre 6 conclut et récapitule ce mémoire en proposant une ouverture sur les limites et pistes de travaux futurs.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre a pour objectif d'offrir un aperçu des méthodes de détection d'anomalies adjacentes au travail effectué. Cette revue de littérature a une structure en entonnoir : on commencera par les méthodes de détection d'anomalie en général, puis on discutera plus spécifiquement du cas des séries temporelles puis enfin, on exposera les propositions de QGANs puisque ces derniers sont au cœur du sujet de recherche. On ne rentrera pas profondément dans les détails techniques et formalisations, ces aspects seront exposés au début du chapitre 3 pour les méthodes retenues.

2.1 Détection d'anomalies

Le problème de la détection d'anomalies est essentiellement une tache de classification avec deux classes : les données normales et les données anormales. Cependant, en pratique, on a un accès facile uniquement à des données normales. On doit donc fabriquer une représentation de ces données normales ou un modèle qui dépend de ces données normales et comparer les données qu'on veut tester à cette construction. Cette section expose les grandes familles de méthodes qu'on retrouve dans la littérature pour résoudre cette tâche.

2.1.1 Méthodes statistiques

La première famille de modèles utilisés pour la détection d'anomalies est celle des modèles probabilistes. Ces modèles cherchent à estimer la distribution de probabilité des données normales et, pour chaque nouvelle donnée à évaluer, de comparer la comparer à la distribution. Si selon la distribution estimée le point est trop peu probable, on le considère comme anormal. L'approche la plus simple dans ce cadre est d'ajuster un modèle statistique comme une gaussienne multivariée aux données d'entraînement et de calculer pour chaque donnée la log-vraisemblance. Cette opération revient à calculer la distance de Mahalanobis [18] entre le point de test et la moyenne des données d'entraînement [19]. Cette méthode a le désavantage de ne s'appliquer qu'aux distributions gaussiennes. Des approches récentes s'inspirent également de ce genre de méthodes pour proposer des versions plus robustes [20]
D'autres propositions permettent de modéliser des distributions plus complexes :

— Un modèle à mixture gaussienne ou Gaussian Mixture Model (GMM) est une généralisation de l'approche précédente. Ces méthodes cherchent à modéliser des distributions possédant des sous populations, chacune suivant une distribution gaussienne [21].

— Un estimateur par noyaux ou Kernel Density Estimator (KDE) qui, en fonction d'une fonction particulière, le noyau ou kernel, construit une distribution de probabilité à partir des données [22]. La densité de probabilité construite par KDE est une somme de kernels prenant en entrée la distance avec chacune des données d'entrainement. Par exemple, pour un kernel uniforme, on obtiendra un histogramme comme distribution de probabilité. En particulier KDE a l'avantage d'être une méthode non paramétrique. KDE a également été le sujet de propositions plus récentes apportant des améliorations [23].

Si avec les améliorations récentes, ces méthodes obtiennent de plutôt bons résultats pour des données de faible dimension, elles souffrent de la malédiction de la dimensionnalité : le nombre de données requises pour obtenir une précision donnée augmente exponentiellement avec le nombre de dimension. Ces méthodes ont besoin d'explorer l'espace de manière approfondie pour être efficaces.

2.1.2 Apprentissage automatique

Depuis l'avènement des modèles d'apprentissage machine et d'apprentissage profond, une grande partie de l'état de l'art dans le traitement de données utilise ce genre de techniques. La détection d'anomalies ne fait pas exception.

Extraction de caractéristiques

Une première approche d'utilisation de l'apprentissage machine pour la détection d'anomalies est l'extraction de caractéristiques [9]. Cela fait écho aux limites des méthodes présentées dans la section précédentes puisque l'extraction de caractéristiques cherche à réduire la dimensionnalité des données en extrayant les composantes importantes. Une fois la dimension des données réduite, on peut appliquer une méthode au choix de détection d'anomalies efficace pour des données de faible dimension comme une méthode statistique ou même une méthode d'apprentissage machine comme un Support Vector Machine (SVM) [24, 25]. Les deux approches basiques d'extraction de caractéristiques sont la projection aléatoire [26, 27] qui effectue des projections linéaires à l'aide de matrices aléatoires et la Principal Component Analysis (PCA) [28, 29] qui en fonction des corrélations entre variables, extrait les sous espaces (ou composantes) dans lesquelles les points du jeu de données sont les plus différents.

Ces méthodes ont toutefois été dépassées par des méthodes plus performantes basées sur l'apprentissage profond. En effet, en s'inspirant de méthodes de classification de l'apprentissage

profond, on peut utiliser des réseaux de neurones pour réduire la dimension de données.
Une première approche est de faire de l'apprentissage par transfert en réutilisant un modèle entraîné sur des données de nature similaire de manière globale [24, 30]. Le modèle général a déjà appris des représentations de données de toutes sortes de classes de données. De cette manière, on contourne le problème de ne pas connaître les données anormales en utilisant un modèle généraliste qui connaît toutes sortes de classes de données.
Si on n'a pas accès à de tels modèles pour les données qu'on veut traiter, on peut également entraîner un modèle d'extractions de caractéristiques dédié aux données. De telles approches sont par exemple basées sur des structures telles que les Auto Encoder (AE) [25, 31]. Les AEs sont des réseaux de neurones profonds avec une sortie de même dimension que l'entrée mais comportant un goulet d'étranglement au milieu en termes de dimension. On entraîne ce type de réseau avec l'objectif de réaliser l'identité : on veut que la sortie soit égale à l'entrée. Seulement, le vecteur en sortie est construit uniquement avec l'information contenue dans le vecteur au goulet d'étranglement qui est un vecteur de dimension inférieur. Une fois l'entraînement fini, on élague le réseau au niveau du goulet pour récupérer une représentation de plus petite dimension des données d'entrée [32]

Les méthodes d'extraction de caractéristiques et de réduction de la dimension des données souffrent cependant du fait que les processus d'extraction et de détection d'anomalies soient étrangers l'un à l'autre : on peut facilement se retrouver avec des bonnes représentations en faible dimension en général, mais pas spécialisées pour la tache de détection d'anomalies pour lesquelles on veut les utiliser.

Approches par reconstruction

Une autre idée exploitant les AEs est d'effectuer l'entraînement uniquement sur des données normales pour apprendre une représentation dans un espace de dimension réduite. L'idée est que si on diminue assez la dimension, la compression effectuée par l'AE doit exploiter les régularités statistiques de la distribution spécifique des données normales au-delà de celles liées au type de données. Si maintenant, on fait passer dans l'AE entraîne une donnée ne suivant pas la distribution apprise, la compression et la reconstruction ne vont pas avoir l'effet escompté. C'est-à-dire que plus la donnée en entrée ne possède pas les régularités exploitées par l'AE, plus la donnée en sortie va être différente. On peut donc construire un score d'anomalie en mesurant une distance entre l'entrée et la sortie d'un AE [33]. On parle de méthode par reconstruction. De nombreuses variations autour des AEs existent dans la littérature pour essayer d'obtenir plus de robustesse par divers procédés, comme l'utilisation

d'un ensemble d'AEs à la place d'un seul [34].

En poussant encore cette idée de détection d'anomalies par reconstruction, une approche est d'utiliser des GANs. Un GAN est un constitué de deux réseaux de neurones, un générateur et un discriminateur. Le discriminateur est entraîné à faire de la classification à une classe et donc à reconnaître une distribution spécifique. Le générateur prend en entrée un vecteur aléatoire d'un espace de faible dimension appelé espace latent et doit générer une donnée que l'on passe dans le discriminateur. On a alors un jeu adversariel : le but du générateur est de faire en sorte que le discriminateur reconnaisse la donnée générée comme appartenant à la distribution qu'il a apprise [35]. La première idée appelée AnoGAN est de rechercher, pour une donnée à tester, le vecteur de l'espace latent qui génère une donnée la plus proche possible [36]. Une fois ce vecteur trouvé, on calcule la distance entre la donnée générée et la donnée d'entrée. Pour une donnée normale, cette distance sera faible et pour une donnée anormale, elle sera plus élevée puisque le générateur est spécialisé dans la génération de données normales. Cependant, ce type d'approche a le défaut de recourir à une recherche une fois l'entraînement terminé, et ce, pour chaque nouvelle donnée à tester. Pour éviter cette recherche, on peut rajouter lors de l'entraînement un réseau chargé d'apprendre l'opération inverse. C'est-à-dire que pour une donnée, on a un agent capable de générer une donnée de l'espace latent qui reconstruit bien la donnée initiale. C'est ce qu'on appelle un GAN bidirectionnel. On se retrouve avec une structure similaire à celle d'un AE. De nombreuses variations de ce type de design ont été proposées notamment en modifiant le design desGANs ou en plaçant la comparaison des données dans l'espace latent [12, 37].
Les approches utilisant des GANs ont comme désavantage principal la difficulté de la convergence du GAN due à sa nature adversarielle. Même quand un équilibre est atteint, des problèmes liés à un générateur trop spécialisé peuvent survenir [38]. Cependant, toutes ces difficultés inhérentes à la nature de la solution sont documentées et possèdent des solutions du fait de la littérature fournie dans le domaine.

2.2 Détection d'anomalies pour les séries temporelles

Après ce tour d'horizon de méthodes de détection d'anomalies générales, cette section discute de l'application de ces méthodes au sous problème spécifique de la détection d'anomalies pour les séries temporelles. Tout d'abord, il s'agit d'éclaircir la notion d'anomalie dans le cadre des séries temporelles. Une anomalie peut être de 3 types différents : des points anormaux, une sous séquence anormale ou encore une série temporelle entière anormale [39]. Les anomalies

vont prendre une de ces formes en fonction du système que l'on enregistre et la durée d'enregistrement choisie. Par exemple, pour des données du MIT BIH arrhythmia Database [40] qui est constitué d'enregistrements de battement de cœurs, on observera des anomalies sous la forme de sous séquences anormales correspondant à des battements de cœur en mauvaise santé. Après un prétraitement qui découpe les séries temporelles de telle manière qu'il y ait un battement par série, on peut transformer ces anomalies de sous séquences en anomalies de séries temporelles entières. On peut alors utiliser des méthodes par reconstruction sur la série temporelle entière [41].
Cependant, pour des données, par exemple issues d'enregistrement d'activité réseau, il est beaucoup plus complexe d'utiliser ce genre de méthodes : les données ne sont pas bien uniformisées temporellement. En effet, le début et la fin des séries temporelles ne correspondent à rien de particulier, elles ont des longueurs qui peuvent être vastement différentes. En fait, pour ce genre de données, la valeur d'un point à un temps donné ne dépend que de l'état du système enregistré à cet instant et pas de la valeur intrinsèque du pas de temps. On est en présence de données ultra-corrélées : la distribution d'un point dans ce genre de série temporelle ne dépend presque uniquement que des points voisins (et uniquement des points précédents d'un point de vue causal). En conséquence, une approche très utilisée pour l'analyse de séries temporelles est de considérer des fenêtres temporelles de longueur constante. On peut typiquement alors s'intéresser à la distribution d'un point en fonction des distributions des points dans son voisinage temporel [39].

On peut alors appliquer toutes sortes de méthodes à ces fenêtres temporelles, comme des méthodes par reconstruction [42, 43], des méthodes statistiques [44, 45], des méthodes spécifiques à des heuristiques sur les données (pas d'apprentissage) ou encore des méthodes par prévision [46–48]. Ces dernières entraînent des modèles à générer le ou les prochains points de la série temporelle. On remplace la tache de classification par une tache de prédiction. Une fois la prédiction faite, on compare cette dernière aux vraies données pour obtenir un score d'anomalies. Comme le modèle s'est entrainé à générer des données normales, la prédiction est sensée être éloignée de la vraie donnée si cette dernière correspond à une anomalie. Différentes approches ont été proposées pour concevoir des agents prédicatifs ,parmi elles se trouvent les GANs conditionnels [49].

2.3 Apprentissage quantique et QGANs

À la suite des premières propositions en apprentissage machine quantique [50, 51], des approches tentant de reproduire des structures classiques sont apparues dans la littérature.

Parmi elles se trouvent les GANs quantiques [14–17, 52]. L'apprentissage machine quantique est simplement l'optimisation classique de systèmes paramétriques quantiques pour réaliser certaines tâches. Lloyd et WeedBrook définissent dans ce cadre une version quantique du jeu adversariel entre générateur et discriminateur [13]. Le principe reste le même qu'avec un GAN classique, cependant les agents paramétriques ne sont pas des réseaux de neurones mais des circuits quantiques variationnels. La majeure différence est donc la nature des agents et pas le déroulé de l'algorithme d'apprentissage. L'optimisation est réalisée à l'aide de méthode de descente de gradients puisqu'il est assez aisé d'accéder aux dérivées partielles des circuits quantiques paramétriques [53, 54]. On peut donc utiliser des approches hybrides avec un générateur quantique et un discriminateur classique ou l'inverse ou encore utiliser deux agents quantiques.

On distingue cependant un type d'approche récent complètement quantique : les QGANs à fidélité d'état [15–17]. Dans ces derniers, générateur et discriminateur sont tous les deux inscrits dans un seul gros circuit quantique. Ces approches ont été récemment privilégiées en raison de meilleures garanties de convergences liées au fait que certaines parties de l'algorithme adversariel soient effectuées au niveau quantique et non classique.

CHAPITRE 3 CONNAISSANCES REQUISES

Ce chapitre a pour objectif d'exposer en détails les connaissances nécessaires pour comprendre la proposition. Il est donc composé d'une section apportant les bases utiles en informatique quantique, puis d'une exposition et formalisation de certains travaux issus de la littérature que la proposition utilise. Enfin, une petite présentation du jeu de données utilisé pour les expérimentations est donnée.

3.1 Connaissances nécessaires en informatique quantique

Cette section a pour but de donner les notions de bases en informatique quantique de base nécessaires pour suivre le travail expliqué plus loin. Il est cependant recommandé pour une compréhension profonde du sujet de consulter un ouvrage plus détaillé comme Quantum Computation and Quantum Information [55].

L'état quantique et la mesure

Contrairement à l'informatique dite classique, l'informatique quantique ne se base pas sur le bit, l'unité basique sur laquelle les opérations sont réalisées est le bit quantique ou qubit. Un qubit est matérialisé physiquement par une propriété binaire d'un système physique quantique, cela peut être la polarisation d'un photon ou le spin d'un électron par exemple. Ici, on se contentera d'une définition théorique d'un qubit, c'est-à-dire une description du modèle mathématique qui régit l'informatique quantique. Un qubit est un vecteur dans un espace vectoriel de Hilbert de dimension 2 noté $\mathcal{H}_2$. C'est-à-dire un espace vectoriel sur le corps des complexes $\mathbb{C}$ muni d'un produit hilbertien $\langle .|. \rangle$ Qui pour rappel est une opération semilinéaire par rapport à la première variable ($\langle aX + X'|Y\rangle = \bar{a}\langle X|Y\rangle + \langle X'|Y\rangle$), linéaire par rapport à la seconde ($\langle X|aY + Y'\rangle = a\langle X|Y\rangle + \langle X'|Y\rangle$) et hermitienne définie positive ($\langle X|X\rangle \in \mathbb{R}_{>0}$). Dans ce contexte, on a donc un qubit φ ou plus généralement $|varphi\rangle$ avec la notation de Dirac s'exprime donc comme suit :

$$|\varphi\rangle = a\,|0\rangle + b\,|1\rangle \tag{3.1}$$

Où a et b sont des scalaires complexes et où $(|0\rangle, |1\rangle)$ constitue une base orthonormale privilégiée appelée la base computationnelle. À partir de cette base computationnelle, on

définit l'opération de mesure comme une variable aléatoire suivant une loi de Bernoulli :

$$M(|\varphi\rangle) \sim \mathcal{B}(|b|^2)$$

Quand on mesure $|\varphi\rangle$ on obtiendra donc 1 avec probabilité $|b|^2$ et 0 avec probabilité $1 - |b|^2$. Pour que cela corresponde à une loi de probabilité, on doit imposer la normalisation de $|\varphi\rangle$ et donc $|a|^2 + |b|^2 = 1$.

Pour manipuler un qubit, on utilise des opérations linéaires sur $\mathcal{H}_2$ avec la seule contrainte de préserver la normalisation des qubits pour que la mesure fasse sens. Les seules applications autorisées sont donc les unitaires. $U \in \mathcal{L}(\mathcal{H}_2)$ est dite unitaire si $UU^\dagger = I_2$ où $U^\dagger$ est la transposée conjuguée de U.

Ce processus se généralise à n qubits en considérant l'espace produit tensoriel $\mathcal{H}_2^{\otimes n}$, dans ce cas, on a un espace à 2^n dimensions muni de la base computationnelle $\{|k\rangle\}_{0 \leq k \leq n}$ avec les entiers écrits en binaires. Un vecteur ou état quantique se décompose alors dans la base computationnelle :

$$|\varphi\rangle = \sum_{0 \leq k \leq n} \varphi_k \, |k\rangle$$

De la même manière, les vecteurs sont normalisés et la mesure se définit comme une variable aléatoire avec la distribution de probabilité suivante :

$$P(M(|\varphi\rangle) = k) = |\varphi_k|^2 \tag{3.2}$$

De même, les opérations sur un état quantiques à n qubits sont exactement les unitaires de $\mathcal{H}_2^{\otimes n}$.

Circuits quantiques et portes usuelles

Une fois ces notions basiques comprises, on présente des opérations unitaires basiques ainsi que comment les assembler pour créer des algorithmes quantiques que l'on représente sous forme de circuits. On appelle aussi les unitaires "portes" par analogie avec les circuits logiques classiques.

Voici quelques exemples d'unitaires usuelles et utilisées dans ce travail :

— La porte d'Hadamard est une unitaire agissant sur un qubit qui transforme la base computationnelle en une base ou les états $|0\rangle$ et $|1\rangle$ sont en superposition équiprobable.

$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

— La porte X est une des trois portes dites de Pauli, elle agit sur un qubit et inverse les composantes selon $|0\rangle$ et selon $|1\rangle$. Elle est aussi appelée porte NOT au vu de son action sur la base computationnelle ($X|0\rangle = |1\rangle$). $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

— La porte Y est la deuxième porte de Pauli, elle agit sur un qubit. $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

— La porte Z est la troisième porte de Pauli, elle agit sur un qubit et change de signe la composante selon $|1\rangle$ du qubit.$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

— Pour toute unitaire, on peut construire la porte U-contrôlée ou $C-U$ qui s'exprime avec la matrice par blocs suivante : $C-U = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$. Le sous-espace correspondant au premier qubit de l'état quantique égal à $|0\rangle$ est inchangé (on applique l'identité). Par contre, pour la composante correspondant au premier qubit égal à $|1\rangle$, on applique U sur les qubits restants.

— La porte SWAP échange les valeurs de deux qubits, sa représentation matricielle est : $SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Pour construire un algorithme quantique, on effectue simplement des transformations unitaires successivement sur un état quantique, comme le produit de deux unitaires est encore une unitaire, la transformation globale est bien valide. Pour représenter ces transformations successives, on utilise la représentation de circuit quantique dont on peut voir un exemple en figure 3.1.

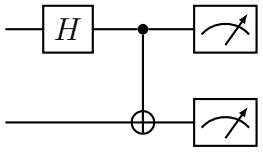


FIGURE 3.1 Un exemple de circuit quantique sur 2 qubits avec des portes usuelles

Chaque fil correspond à un qubit, quand on a deux portes en parallèle U et V cela correspond à l'unitaire globale $U \otimes V$, donc ici, on a par exemple une porte d'Hadamard en parallèle

avec un fil, l'unitaire à deux qubits s'exprime donc $H \otimes I = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$ puisque le fil correspond à l'identité. Si on applique une unitaire U sur le qubit i, on note l'unitaire globale sur les n qubits correspondante U_i. Dans le cas de l'exemple, on a donc $H_1 = H \otimes I$ Les symboles en fin de circuit correspondent aux mesures sur les qubits.

Sphère de Bloch et portes paramétriques

Une représentation utile d'un qubit est celle de la sphère de Bloch, visible dans la figure 3.2. Cette représentation provient de la décomposition d'un qubit suivante (à une phase globale près, invisible dans la mesure) :

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{3.3}$$

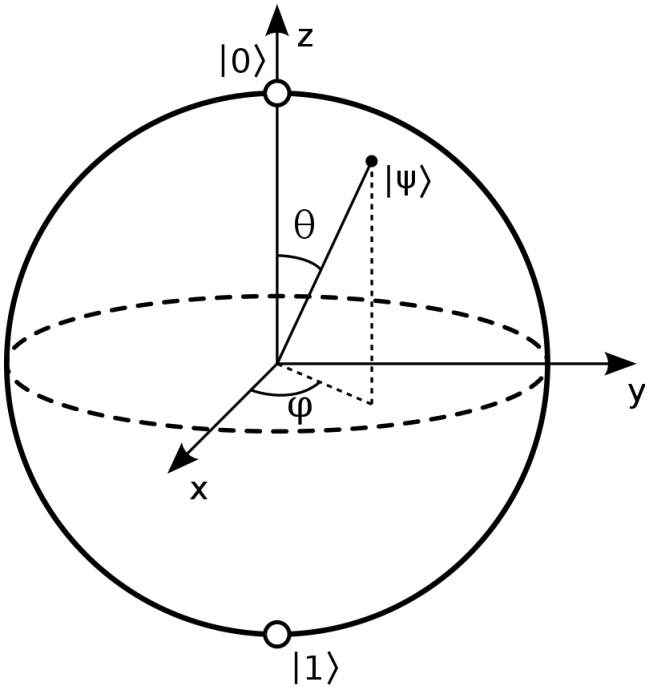


FIGURE 3.2 Sphère de Bloch par Smite-Meister - Own work, CC BY-SA 3.0, https ://commons.wikimedia.org/w/index.php ?curid=5829358

Dans cette représentation, les unitaires appliquées à un qubit sont toutes des rotations sur la sphère autour d'un axe. En particulier, on peut définir les rotations autour de 3 axes

spécifiques x, y et z.

$$R_Y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{3.4}$$

$$R_X(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ \sin\left(-i\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \tag{3.5}$$

$$R_Z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{3.6}$$

On peut, en combinant ces trois rotations paramétrées, fabriquer n'importe quelle transformation unitaire sur un qubit.

3.2 Formalisations

Cette section vise à présenter de manière formelle les constructions théoriques utiles à la proposition finale et d'expliquer leur fonctionnement.

3.3 QGAN

Un GAN est défini comme un jeu adversariel de la manière suivante [35] : pour une distribution μ_{ref} de référence sur un espace Ω, on a deux agents, le générateur et le discriminateur. Le générateur a pour ensemble de stratégies l'ensemble de toutes les distributions sûr Ω. Le discriminateur quant à lui peut jouer l'ensemble des applications f_D qui, à un élément de Ω, associent une distribution de probabilité sur $[0, 1]$. On définit alors un jeu à somme nulle avec la fonction-objectif suivante :

$$L(\mu_G, f_D) = \mathbb{E}_{x\sim\mu_{ref}, y\sim f_D(x)}[ln(y)] + \mathbb{E}_{x\sim\mu_G, y\sim f_D(x)}[ln(1-y)] \tag{3.7}$$

Cette fonction est la fonction de gain du discriminateur qu'il cherche donc à maximiser, celle du générateur est son opposée.

Qualitativement, le discriminateur s'entraîne sur la distribution de référence et essaye d'apprendre à reconnaître les données suivant cette distribution, ce qui est représenté par le premier terme de 3.7. D'autre part, le générateur essaye de générer des données qui vont tromper le discriminateur. Son but est de faire en sorte que la donnée générée soit reconnue par le discriminateur comme faisant partie de la distribution de référence. Le discriminateur à

l'opposé a pour objectif de ne pas se faire tromper par le générateur. Cette confrontation entre les deux agents est transcrite dans le deuxième terme de 3.7. Une description schématique du fonctionnement d'un GAN est visible en figure 3.3.

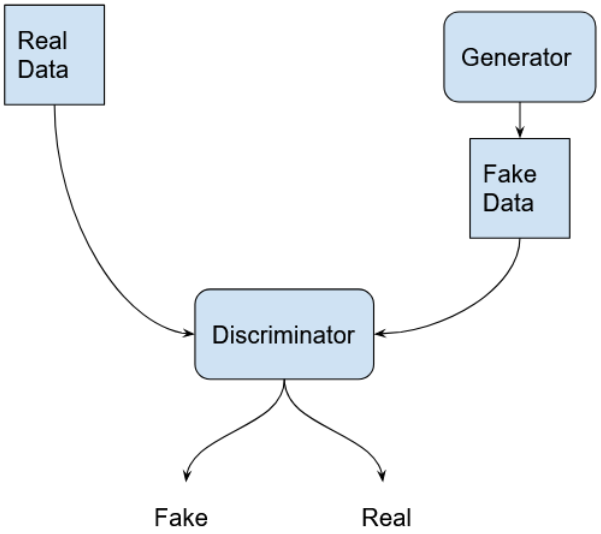


FIGURE 3.3 Schéma du fonctionnement d'un GAN

En pratique, on entraine successivement le discriminateur contre les données réelles puis le générateur et le discriminateur l'un contre l'autre, le générateur prend également en entrée un vecteur aléatoire de petite dimension comme une graine de génération pour qu'il soit capable de fabriquer des résultats différents. L'implémentation pratique utilise pour le générateur et le discriminateur des agents paramétriques, typiquement des réseaux de neurones. Dans notre cas, puisqu'on est dans un cadre quantique, on utilisera leur équivalent quantique : les QVCs.

3.3.1 Circuits variationnels quantiques

Un circuit variationnel quantique est simplement un circuit quantique possédant des portes paramétriques [56]. Si on veut que la fonction réalisée par le circuit prenne des données classiques en argument (ce qui est le cas dans le cadre de ce travail), on effectue un encodage des données en plaçant devant le circuit variationnel. L'encodage par angle est la méthode d'encodage privilégiée dans la littérature du fait de sa simplicité et du peu de ressources qu'elle nécessite comparé à d'autres méthodes comme l'encodage par amplitude. L'encodage par angle consiste à encoder une donnée numérique sur chaque qubit en effectuant une rotation proportionnelle à la donnée. Un exemple de QVC avec encodage par angle est visible en

figure 3.4. Dans ce cas, l'encodage par angles est effectué avec des portes R_Y.

Concernant les sorties de QVCs on peut considérer qu'au niveau le plus fondamental, la sortie est l'état quantique disponible en fin de circuit. Cependant, si on est dans un contexte ou les données d'origine sont classiques, on doit se poser la question de l'interprétation de cet état quantique pour récupérer une donnée classique. Une approche peut être issue du fait d'estimer la distribution de probabilité définie par l'état quantique (3.2), en mesurant un grand nombre de fois l'état quantique de sortie du circuit, on peut reconstruire la distribution de probabilité définie par les coefficients de l'état quantique dans la base computationnelle. On récupère alors un vecteur normalisé de dimension 2^n si on travaille avec n qubits, c'est-à-dire une distribution de probabilité sur $0, ..., 2^n$. On peut alors interpréter cette distribution de probabilité en effectuant une transformation vers un autre espace ou alors simplement interpréter les valeurs du vecteur de sortie de la manière dont on veut. En fonction de l'interprétation, on peut construire une fonction de coût de son choix.

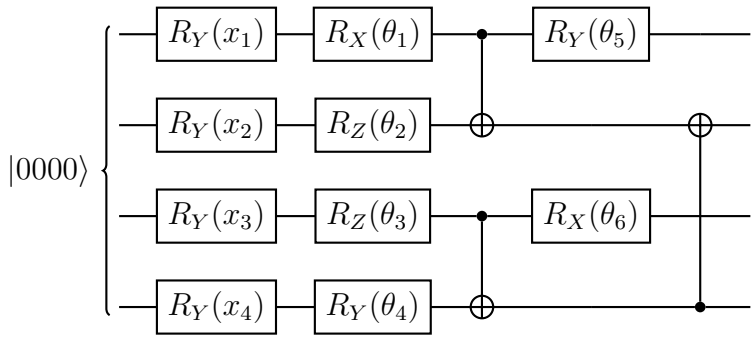


FIGURE 3.4 Exemple de circuit variationnel quantique

Il reste tout de même le problème de l'optimisation des paramètres pour effectuer la phase d'apprentissage. L'approche de choix est d'utiliser un algorithme d'optimisation classique, le plus souvent par descente de gradient comme en apprentissage machine classique [56]. Cela est rendu possible par la possibilité de calculer des dérivées partielles de circuit quantiques par rapport aux paramètres des portes les composant. Pour cela, on utilise des règles dites de décalage de paramètre (ou parameter shift) [53, 54]. Pour les portes à un qubit, la règle est décrite par l'équation (3.8). Ici, f est une fonction objectif dépendant de l'état quantique en sortie du circuit.

$$\frac{\delta f}{\delta \theta} = \frac{1}{2}(f(\theta + \frac{\pi}{2}) - f(\theta - \frac{\pi}{2})) \tag{3.8}$$

Concernant le design d'un QVC, il y a deux métriques principales à prendre en compte : l'expressibilité et la capacité d'intrication. L'expressibilité est la mesure de la capacité d'un circuit paramétrique à explorer une grande partie de l'état quantique en modifiant ses paramètres. On la mesure typiquement en comparant les sorties du QVC avec la distribution maximalement uniforme dite de Haar [56, 57]. On peut également quantifier la capacité d'intrication. Cette dernière est, comme son nom l'indique, la capacité du circuit à créer des états quantiques intriqués.

3.3.2 GANs quantiques

Une fois les deux notions de GAN et de QVC établies, on peut définir le QGAN. Comme la version classique, le QGAN est défini comme un jeu à somme nulle [13]. Il suit les mêmes règles que le GAN classique, mais les implémentations du générateur et discriminateur impliquent des QVCs. De même, l'espace Ω peut être soit un espace de données classiques, soit un espace quantique. S'il existe de nombreuses approches hybrides, celle choisie est une approche complètement quantique appelée "state fidelity QGAN" [15–17].

SWAP test

Cette approche repose sur un circuit qui effectue une fonction appelée "SWAP test" qui en fonction de deux états quantiques $|\varphi\rangle$ et $|\psi\rangle$ renvoie sur un qubit auxiliaire un état qui porte une information sur la similarité entre $|\varphi\rangle$ et $|\psi\rangle$. Ce sous-circuit est visible en figure 3.5.

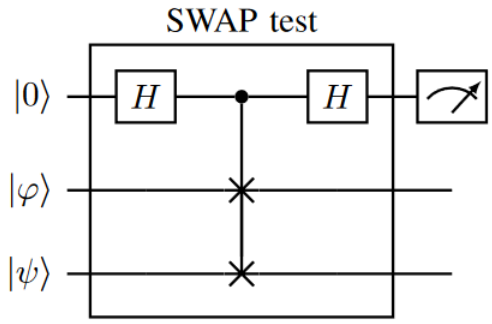


FIGURE 3.5 Circuit quantique du SWAP-test

L'état quantique est initialement de la forme $|0\rangle \otimes |\varphi\rangle \otimes |\psi\rangle$ aussi noté $|0, \varphi, \psi\rangle$. Après la porte de Hadamard sur le premier qubit, on obtient l'état décrit par (3.11). La porte suivante est un SWAP contrôlé qui échange donc les deux états $|\varphi\rangle$ et $|\psi\rangle$ pour la composante du premier

qubit selon l'état $|1\rangle$. On obtient, après cette étape, l'état visible en (3.12). Enfin on réapplique une porte de Hadamard sur le premier qubit et on obtient (3.15).

$$H_1 |0, \varphi, \psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |\varphi\rangle \otimes |\psi\rangle \tag{3.9}$$

$$= \frac{1}{\sqrt{2}}(|0, \varphi, \psi\rangle + |1, \varphi, \psi\rangle) \tag{3.10}$$

$$\text{C-SWAP}(\frac{1}{\sqrt{2}}(|0, \varphi, \psi\rangle + |1, \varphi, \psi\rangle) = \frac{1}{\sqrt{2}}(\text{C-SWAP}(|0, \varphi, \psi\rangle) + \text{C-SWAP}(|1, \varphi, \psi\rangle)) \tag{3.11}$$

$$= \frac{1}{\sqrt{2}}(|0, \varphi, \psi\rangle + |1, \psi, \varphi\rangle) \tag{3.12}$$

$$H_1(\frac{1}{\sqrt{2}}(|0, \varphi, \psi\rangle + |1, \psi, \varphi\rangle)) = \frac{1}{\sqrt{2}}(H_1 |0, \varphi, \psi\rangle + H_1 |1, \psi, \varphi\rangle) \tag{3.13}$$

$$= \frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(|0, \varphi, \psi\rangle + |1, \varphi, \psi\rangle) + \frac{1}{\sqrt{2}}(|0, \psi, \varphi\rangle - |1, \psi, \varphi\rangle)) \tag{3.14}$$

$$= \frac{1}{2} |0\rangle \otimes (|\varphi, \psi\rangle + |\psi, \varphi\rangle) + \frac{1}{2} |1\rangle \otimes (|\varphi, \psi\rangle - |\psi, \varphi\rangle) \tag{3.15}$$

Une fois cet état construit, on effectue une mesure sur le premier qubit, cela se traduit par une trace partielle sur le reste de l'état quantique. Ultimement, la probabilité de mesurer 0 sur le premier qubit est donnée par (3.16)

$$P(\text{Mesure} = 0) = \frac{1}{2}(1 + \langle\varphi|\psi\rangle^2) \tag{3.16}$$

Cette probabilité est appelée "state fidelity" et à partir d'elle, on peut calculer la similarité $\langle\varphi|\phi\rangle^2$ entre les deux états en entrée du SWAP test.

State fidelity QGANs

A partir de ce swap test, on peut construire un design de QGAN suivant les circuits visibles en figures 3.6 et 3.7.

D_θ est le circuit du discriminateur quantique et $G\theta$ est le circuit du générateur quantique. Le circuit en figure 3.6 correspond à la situation où le discriminateur est entraîné contre les données réelles. L'autre circuit en figure 3.7 correspond à la situation où les deux agents s'entraînent l'un contre l'autre. Les deux agents sont implémentés par des QVCs. On peut remarquer que dans ce cas, les circuits du discriminateur et du générateur ont des rôles sy-

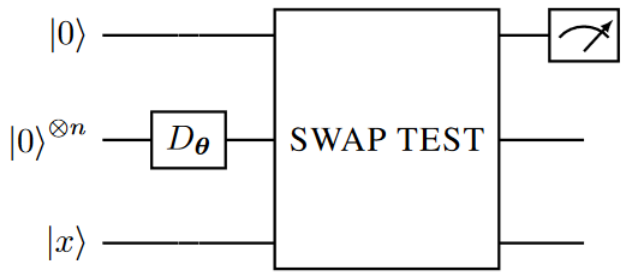


FIGURE 3.6 Circuit d'un QGAN à fidélité d'état dans la configuration discriminateur contre données réelles

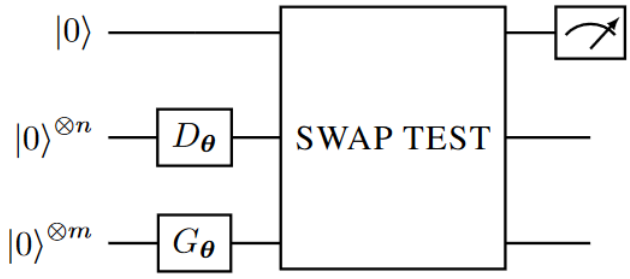


FIGURE 3.7 Circuit d'un QGAN à fidélité d'état dans la configuration discriminateur contre générateur

métriques puisque le SWAP test ne différencie pas entre les deux états d'entrée. En effet, la chose qui différencie les deux agents dans ce paradigme est simplement leur rôle dans le jeu adversariel.

Cette approche offre un avantage de convergence par rapport aux méthodes où on remplace simplement les agents classiques par des agents quantiques dans l'architecture GAN. En effet, si on fabrique des fonctions de pertes à partir des sorties d'agents quantiques, on rencontre le problème que plusieurs états quantiques correspondent à la même distribution en sortie. En construisant une fonction objectif à partir de la distribution en sortie d'un QVC, on obtient une influence sur les probabilités de cette distribution et donc sur le module des coefficients de l'état quantique de sortie. Mais plusieurs nombres complexes correspondent à une même amplitude et donc plusieurs circuits variationnels quantiques correspondent à une même distribution. On peut alors se retrouver avec des instabilités dans la convergence des paramètres des QVC. Dans le cadre des QGANs basés sur le SWAP test, on mesure directement la fidélité entre les deux états quantiques comparés et on construit les fonctions objectifs à partir

de cette mesure nous renseignant sur les sorties des QVCs directement au niveau de l'état quantique. Ces fonctions de perte sont décrites par les équations (3.17) pour la fonction de perte du discriminateur contre les données réelles, (3.18) et (3.19).

$$L_{D,X}(x) = -log(E[\langle\varphi|x\rangle]) \tag{3.17}$$

$$L_{D,G} = -log(1 - E[\langle\varphi|\psi\rangle]) \tag{3.18}$$

$$L_{G,D} = -log(E[\langle\varphi|\psi\rangle]) \tag{3.19}$$

On entraîne donc le discriminateur en comparant l'état quantique en sortie du discriminateur $|\varphi\rangle$ avec un état quantique encodant une donnée réelle $|x\rangle$ et en optimisant en fonction de l'espérance de la similarité entre les deux états : si ils sont similaires $\langle\varphi|x\rangle$ est proche de 1 et donc $L_{D,X}(x)$ est proche de 0 et plus les états sont orthogonaux plus $\langle\varphi|x\rangle$ est proche de 0 et plus $L_{D,X}(x)$ est grand. D'une manière similaire, on entraîne le générateur et le discriminateur l'un contre l'autre en récompensant le discriminateur avec $L_{D,G}$ proche de 0 si son état en sortie $|\varphi\rangle$ est proche de celui en sortie du générateur $|\psi\rangle$ et en récompensant le générateur avec $L_{G,D}$ proche de 0 si les deux états sont éloignés.

3.4 Données et contraintes

Ce travail de recherche a initialement été pensé dans le cadre de l'utilisation de données réseaux. En effet, dans le cadre de l'activité d'un réseau informatique, on enregistre de nombreux paramètres qui décrivent l'évolution du système. De nombreuses bases de données existent, mais ne sont pas toutes adaptées directement aux séries temporelles. En général, les jeux de données d'anomalies réseau sont des enregistrements d'événements qui comportent de nombreuses caractéristiques et on détecte alors des anomalies en fonction de ces caractéristiques [58]. On peut adapter ce type de jeux de données basés sur les événements en série temporelle à condition que les événements soient horodatés. En échantillonnant avec un pas de temps inférieur à la moitié plus petit écart temporel entre deux événements, on se retrouve avec une série temporelle. Cette série temporelle est à priori multivariée et contient de nombreuses composantes catégoriques. Il faut alors sans doute, dépendamment de la méthode qu'on utilise, réaliser un prétraitement pour numériser ces composantes voire réduire le nombre de composantes du signal. C'est particulièrement vrai dans notre cas puisque les contraintes de l'informatique quantique sont beaucoup plus restrictives que dans le cas classique. Dans le cadre du développement de ce premier système de détection d'anomalies avec

un système quantique, on souhaite se limiter à des signaux univariés numériques si possible.

Le jeu de données utilisé pour ce travail est le jeu de données AWSCloudwatch du Numenta Anomaly Benchmark. La démarche de ce benchmark est de proposer des jeux de données pour l'évaluation de la détection d'anomalie pour les séries temporelles. Plus spécifiquement, le jeu de données AWSCloudwatch contient des données issues de serveurs amazon web services. Il est constitué de séries temporelles enregistrant au cours du temps l'évolution d'indicateur clés de performance comme l'activité du processeur d'un serveur ou le nombre de requêtes au serveur. En plus de ces séries temporelles, NAB fournit des étiquetages des instants auxquels des anomalies sont survenues ainsi que des fenêtres temporelles autour de ces instants qui donnent les zones de détection valides. Si un système de détection d'anomalies signale une anomalie à l'intérieur de ces fenêtres, on doit considérer que c'est un vrai positif. Inversement, si on détecte une anomalie en dehors de ces fenêtres, le système a fait une erreur [59]. On voit en figure 3.8 un exemple d'une série temporelle de AWSCloudwatch avec les anomalies et les fenêtres temporelles marquées en rouge.

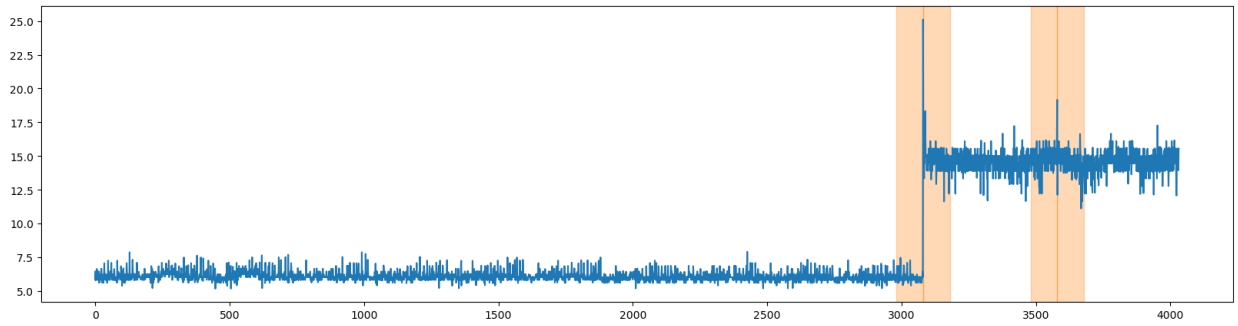


FIGURE 3.8 Exemple de série temporelle représentant l'activité d'un processeur sur 4000 pas de temps.

Le reste des séries temporelles utilisées est disponible en annexe en figures A.1 et A.2.

CHAPITRE 4 DÉMARCHE ET ORGANISATION

Dans ce chapitre, on présente la méthodologie de travail globale de ce projet de recherche. On rappelle que le but est de fournir une première solution pour adapter les propositions de la littérature en matière de QGANs au problème de la détection d'anomalies pour les séries temporelles.

4.1 Méthodologie

Le travail s'est divisé en plusieurs étapes :

- Reproduire les diverses propositions de QGANs dans la littérature. Puis choisir une base parmi elles.
- Essayer plusieurs design de QGANs adaptés aux séries temporelles sur des données artificielles pour tester certaines propriétés importantes.
- Sélectionner la meilleure option et construire un système de détection d'anomalies à partir d'elle.
- Tester le système de détection d'anomalies avec sur un jeu de données commun de la littérature en détection d'anomalies pour les séries temporelles

4.2 Code

L'implémentation des QGANs et de tout autre système a été réalisé en python 3 à l'aide des librairies principales suivantes : numppy, pandas pour traiter les données et qiskit la libraire d'informatique quantique développée par IBM. Les tests ont été réalisés sur simulateur quantique dans l'optique d'offrir une preuve de concept sur le design du QGAN proposé. En effet, ce choix permet une beaucoup plus grande flexibilité sur les expérimentations puisque toutes les opérations s'effectuent sur une machine dédiée. Des notebooks de code ont été produits et représentent un livrable pour ce projet.

CHAPITRE 5 ARTICLE 1 : SUCCESSIVE DATA INJECTION IN CONDITIONAL QUANTUM GAN APPLIED TO TIME SERIES ANOMALY DETECTION

labelsec :Theme2

Contribution : J'ai contribué à l'article en concevant la contribution principale, en réalisant les démonstrations expérimentales et itérations de la proposition, en rédigeant la majeure partie du manuscrit et en faisant une partie de la recherche bibliographique. Cet article a été publié le 26 janvier 2024 dans IET Quantum Communication. Les coauteurs sont Soumaya Cherkaoui, Jean-Frédéric Laprade, Shengrui Wang et Ola Ahmad.

abstract : Classical GAN architectures have shown interesting results for solving anomaly detection problems in general and for time series anomalies in particular, such as those arising in communication networks. In recent years, several quantum GAN architectures have been proposed in the literature. When detecting anomalies in time series using QGANs, huge challenges arise due to the limited number of qubits compared to the size of the data. To address these challenges, we propose a new high-dimensional encoding approach, named Successive Data Injection (SuDaI). In this approach, we explore a larger portion of the quantum state than that in the conventional angle encoding, the method used predominantly in the literature, through repeated data injections into the quantum state. SuDaI encoding allows us to adapt the QGAN for anomaly detection with network data of a much higher dimensionality than with the existing known QGANs implementations. In addition, SuDaI encoding applies to other types of high-dimensional time series and can be used in contexts beyond anomaly detection and QGANs, opening up therefore multiple fields of application.

5.1 Introduction

Time series are an omnipresent data type in the scientific and industrial world. Because of their temporal nature, they are widely observed where it is necessary to monitor various variables in systems over time. Examples of the different fields where events occur over time range from finance, to meteorology and networking [6,60,61]. In all of these fields it is crucial to be able to detect when an abnormal behaviour occurs [2, 4, 5]. Thus, anomaly detection in time series has been for decades a very active subjet in data analysis in general [7], and in particular in the field of communication and networks. This is all the more important as networks have become an integral part of the infrastructure of today's connected and

smart society. Although some anomalies can be mere malfunction or external events, many anomalies in networks come from cyberattacks that are growing in sophistication. As a result, anomaly detection has garnered increasing research attention. Given the substantial consequences of a successful attack, there are strong motivations for attackers to improve their attack methods. Conversely, defenders also have significant incentives to enhance their defense strategies. To gain an edge in this adversarial context, it can be strongly advantageous to explore innovative technologies that can help tipping the scales in favor of defenders.

In the last decade, classical machine learning (ML) has proven to be highly effective at analyzing and extracting insight from data. As a result, ML has become the primary contributor in the advancement in the field of anomaly detection, with many of the state-of-the-art propositions using deep learning methods [9]. Among these ML techniques, generative methods, more specifically Generative Adversarial Networks or GANs, have shown interesting results [62]. For instance, the early work [36] on GANs shows that one can detect anomalies by comparing the outputs of the generative model with real data. Newer approaches extend this concept by adding the utilisation of the encoder/decoder structure found in bidirectional GANs. This enables a more efficient data comparison by employing the latent space output of the encoding process [12,37]. In the context of network time series, because of the large data size, it is common to use a sliding time window to reduce the size of the data to be processed. Models must then take into account the contextual data contained within the window. One way to achieve this is to work on the conditional distribution of future data as a function of previous data. For this purpose, a conditional GAN or cGAN is practical, this type of GAN learns to replicate conditional distributions by taking contextual data as an input [63].

Meanwhile, with the recent progress in quantum hardware [64], we have started to see experiments that demonstrate a quantum advantage over known classical methods [8,65]. There have been propositions in the field of quantum machine learning drawing inspiration from classical techniques [50, 51, 56], time series more specifically have been studied through the quantum machine learning scope in recent years, notably by using quantum reservoirs [66,67]. In particular, several quantum GAN (QGAN) architectures have been proposed since the founding article [13] in 2018 [14, 52]. Recently, a new kind of QGAN has been developed in several papers : state fidelity based QGANs [15–17]. These offer increased convergence properties when compared to the previous propositions by using a quantum state comparison sub-circuit.

Given the demonstrated success of GANs in detecting anomalies and the ongoing progress in QGANs, the prospect of applying QGANs to anomaly detection is becoming an interesting and promising research avenue. However, all the existing QGAN approaches in the literature suffer from the drawback of not putting much emphasis on data encoding. These approaches do not address the specific case of a conditional Quantum Generative Adversarial Network (cQGAN). In fact, when attempting to directly adapt these QGAN architectures for anomaly detection and transform them into cQGANs, we encounter limitations related to the number of qubits. The issue lies in the fact that almost all QGAN architectures described in the current literature rely on angle encoding to represent real data. This encoding requires a number of qubits that scales lineary with the size of the encoded data. For time series data, where a substantial contextual window of data is used, this encoding quickly becomes impractical. Consequently, our work tackles the problem of making anomaly detection on time series using a quantum GAN practical in a scalable manner.

In this paper, we propose a novel approach to tailoring existing QGAN techniques for time series and their high dimensionality. Our main contribution is the design of a novel scheme, called Successive Data Injection (SuDaI), to allow for an encoding of high dimensional data point into a quantum state sequentially. We then demonstrate its effectiveness in a quantum design by building an anomaly detection system and testing it on a network anomaly detection dataset that we preprocess into time series data.

This article's structure is as follows : section 5.2 provides the necessary background for understanding the proposition by presenting quantum computing basics, QGANs and the constraints and hypothesis for our specific application that lead to the proposition. Section 5.3 exposes the design of the proposed quantum encoding method and of the conditional quantum GAN used. Section 5.4 details the functioning of the anomaly detection system built around the QGAN proposition. Section 5.5 presents the details of our implementations including choice of the dataset, experiment description and an analysis of the results of the QGAN and of the anomaly detection system. Section 5.6 summarizes the contributions and the limitations of this work and suggests possible paths for future improvements.

5.2 Background

5.2.1 Quantum Background

The unit of information of quantum computation [55] is called the qubit (for quantum bit). Mathematically, a qubit is a vector in a Hilbert space of dimension 2, $\mathcal{H}_2$. One special orthonormal basis of this vector space is noted as $\{\vec{0}, \vec{1}\}$ or more commonly $\{|0\rangle, |1\rangle\}$ using Dirac notation. This basis is often named the computational basis and one can then express a qubit $|\varphi\rangle$ in this basis. Let $(\alpha, \beta) \in \mathbb{C}^2$ then,

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle . \tag{5.1}$$

Another constraint on qubits is that they should be normalized, meaning $||\varphi||^2 = |\alpha|^2 + |\beta|^2 = 1$. This is necessary to define an operation on qubits called a measurement. Measuring a qubit $|\varphi\rangle$ defined as in (5.1) in the computational basis yields either 0 with probability $|\alpha|^2$ or 1 with probability $|\beta|^2$. The normalization ensures that the measurement samples from a proper probability distribution.
To manipulate a qubit, any linear transformation that preserves this normalization constraint is allowed. This constitutes the space of unitary operations of $\mathcal{H}_2$. U is said to be unitary if and only if $UU^\dagger = U^\dagger U = I$ where $U^\dagger$ is the conjugate transpose of U.

These rules are generalized for n qubits by considering the n^{th} power of $\mathcal{H}_2$ with the tensor product, i.e. $\mathcal{H}_2^{\otimes n}$, which is a 2^n dimensional Hilbert space. The computational basis in which the quantum states can be expressed (Eq. (5.2)) is then $\{|k\rangle\}_{0 \leq k < 2^n}$, with k being a binary number. A measure yields one of those states with a probability equal to the square of the corresponding amplitude modulus (Eq.(5.3)). Allowed operations are the unitary operations on $\mathcal{H}_2^{\otimes n}$.

$$|\varphi\rangle = \sum_{k<2^n} \varphi_k |k\rangle \tag{5.2}$$

$$P(Measure = k) = |\varphi_k|^2 \tag{5.3}$$

From this point, one can manipulate a system of qubits (also called a quantum state) by applying successive unitary operations to it. The common representation of quantum algorithms is a quantum circuit (see Fig. 5.1).

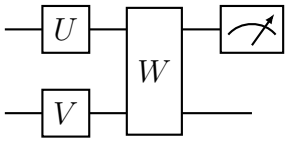


FIGURE 5.1 An example of a quantum circuit on two qubits with unitaries U, V, W and a measurement on the first qubit.

In this example circuit, each line represents a single qubit, a box is a unitary (or quantum gate in this context) that is applied to all the qubits that pass through the box. Following this convention, W is a two-qubits gate. When two unitaries U and V are applied in parallel to different qubits, it is equivalent to the unitary $U \otimes V$ applied on the whole quantum state.

5.2.2 GANs and Quantum GANs

Generative adversarial networks consist of 2 agents : the discriminator and the generator [35]. The discriminator is a classifier trained on a given real distribution. The generator produces fake data with the goal of fooling the discriminator into classifying it as part of the real distribution it learns to recognize. The discriminator is thus also trained to recognize the data of the generator as not being part of the real distribution. At the end of this adversarial game, the generator should generate data according to the real distribution. A schematic describing the pipeline of a GAN is shown in Fig. 5.2.

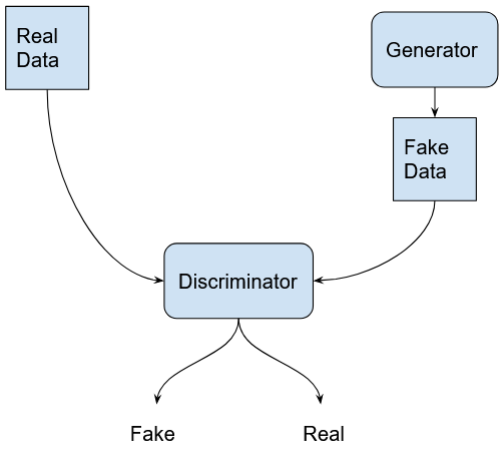


FIGURE 5.2 A schematic of the pipeline of a GAN.

These discriminator and generator are typically implemented as neural networks and are optimized through gradient descent methods. Analogous agents in quantum computing have

already been proposed in the literature, such as Quantum Variational Circuits (QVCs) [51,56]. They consist of quantum circuits with parametric gates, the parameters are then optimized with classical techniques using gradient descent with respect to an objective function. To be able to perform such optimization, one needs to access the derivatives of the circuit, which are obtainable through the parameter shift rule [53, 54].

In this work, we will be using $\boldsymbol{\theta}$ to represent the vector containing all the parameters used in a given circuit, θ or θ_j will be used to represent a single real parameter. Let l be a loss function for our quantum variational circuit that depends on a real parameter $\theta_i \in \boldsymbol{\theta}$, then

$$\frac{\mathrm{d}l}{\mathrm{d}\theta_i} = \frac{1}{2}\left(l\left(\boldsymbol{\theta} + \frac{\pi}{2}\mathbb{1}_i\right) - l\left(\boldsymbol{\theta} - \frac{\pi}{2}\mathbb{1}_i\right)\right), \tag{5.4}$$

where $\mathbb{1}_i$ is a vector that has the same number of dimensions as $\boldsymbol{\theta}$ and takes value 1 at index i and 0 otherwise.

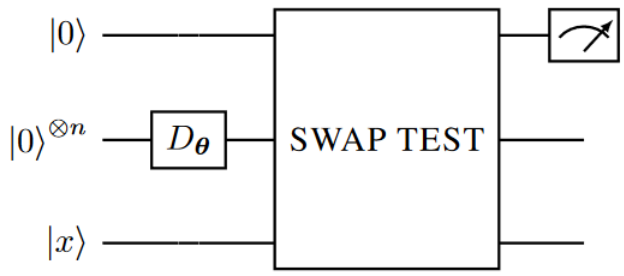
The type of quantum GAN that will be used in this work is one that has been recently developed, namely state fidelity quantum GANs [15–17]. The QGAN is described by the two circuits shown in Fig.5.3a and Fig.5.3b. The $D_{\boldsymbol{\theta}}$ and $G_{\boldsymbol{\theta}}$ unitaries are quantum variational circuits for the discriminator and the generator. $|x\rangle$ is an encoding of a real data point x inside a quantum register.

The component at the end of the two circuits is called the SWAP test, which takes as input two quantum states $|\varphi\rangle$ and $|\psi\rangle$ with the same dimension, and outputs on an auxiliary qubit, a state that encodes the similarity between the two input quantum states [17]. The details of this part of the circuits can be seen in Fig.5.4. When measuring this qubit the output follows the probability given by eq. (5.5). This value is called the fidelity between the states $|\varphi\rangle$ and $|\psi\rangle$. It can be estimated by sampling from the auxiliary qubit.

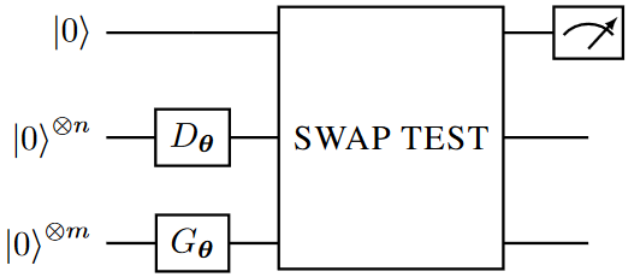
$$P(Measure = 0) = \frac{1}{2}(1 + \langle\varphi|\psi\rangle^2), \tag{5.5}$$

Finally, the similarity $\langle\varphi|\psi\rangle$, which is the Hermitian product between $|\varphi\rangle$ and $|\psi\rangle$, can be extracted and used to construct a loss for the generator and discriminator.

The SWAP test is used inside the QGAN design to compare the output states of the generator and discriminator circuit or the discriminator's output state with the real data representation. The controlled SWAP operation requires both input states to have the same number of



(a) The discriminator $D_{\boldsymbol{\theta}}$ and the real data $|x\rangle$.



(b) The discriminator $D_{\boldsymbol{\theta}}$ and the generator $G_{\boldsymbol{\theta}}$.

FIGURE 5.3 State fidelity QGAN circuit designs.

qubits. In practice we designate sub-registers for the discriminator and the generator. These registers must also have a number of qubits equal to of the state representing the real data. The remaining qubits outside the registers are not used. This means that the generator and discriminator have a number of qubits greater than or equal to that of the state encoding real data. In most cases reported in the literature, all the qubits of the generator and discriminator are used and thus both circuits use the same number of qubits. This number is also the dimension of the data encoded with a basic angle encoding. In our work, as the generated point is one dimensional, the sub registers will be both composed of one qubit.

Note that in this QGAN design, the generator and discriminator have very similar architectures. In fact, in most published cases, the discriminator and generator sub-circuits employs the same architecture. They are distinguished by the different roles played inside the adversarial game. Those roles are determined by their attributed loss functions. For our purpose, they will be respectively : i) $L_{D,X}$, the discriminator loss against real data x (eq. (5.6)), ii) $L_{D,G}$, the discriminator loss against the generator (eq. (5.7) and iii) $L_{G,D}$, the generator loss

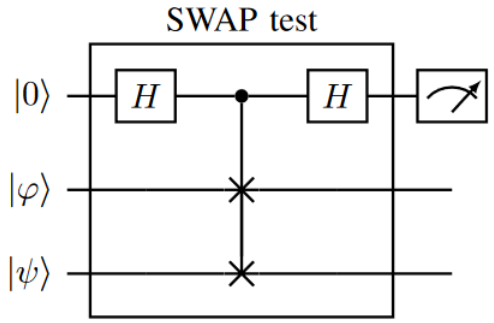


FIGURE 5.4 Circuit of the SWAP test composed of two Hadamard gates and a controlled SWAP gate.

against the discriminator (eq. (5.8)) :

$$L_{D,X}(x) = -log(E[\langle\varphi|x\rangle^2]) \tag{5.6}$$

$$L_{D,G} = -log(1 - E[\langle\varphi|\psi\rangle^2]) \tag{5.7}$$

$$L_{G,D} = -log(E[\langle\varphi|\psi\rangle^2]) \tag{5.8}$$

5.2.3 Limitations in Quantum Data Loading

Recall that we aim to detect anomalies from windowed time series data using a quantum generative model. This type of data is typically high dimensional depending on the number of time steps that may be in the thousands or more. This is challenging given the current constraints on quantum hardware, which is for now limited in terms of number of qubits, connectivity between qubits and coherence times. Even though a quantum state of n qubits could encode an exponential amount of data in its amplitudes, preparing this quantum state has a time and space complexity that makes this approach not viable. The most common data encoding strategy used in quantum variational algorithms is known as angle encoding where n real values are encoded into $O(n)$ qubits by applying a parametrized rotation gate to each qubit. This strategy has the advantage of requiring few computational resources when compared to more information dense methods such as amplitude encoding. However, angle encoding requires a linear number of qubits and does not take advantage of the exponentially large vector space.

5.3 The Proposed Quantum GAN

5.3.1 The Conditional Quantum GAN

To reduce the dimensionality of the data, our approach involves employing a well-established technique commonly used in anomaly detection, consisting in the utilization of a shifting time-window [39]. The core concept behind this method is to identify anomalies within a time series at a specific time step by considering the preceding data points within a defined time window. Instead of dealing with time series data with an unlimited number of time steps, we simplify the problem by working with fixed-length time windows. Given that our problem now encompasses a conditional distribution, we build a slightly modified architecture for the QGAN, namely a conditional QGAN or a cQGAN. Specifically, both the generator and discriminator need to be preceded by an encoder circuit, which encodes the data from the preceding time steps of the time series. In a more general description, we want to learn the probability distribution of B given A. Fig.5.5 describes the pipeline of a conditional GAN [63]. The addition of the contextual data following the distribution A is highlighted in orange when compared to the original GAN pipeline.

The main difference with typical GANs is that both the generator and the discriminator are fed with the contextual data a in order to discriminate or generate the b object. In the case of a sliding time window in a time series, a is the data contained in the time window and b is the next data point. The architecture for the cQGAN then becomes the one shown in Fig.5.6a and Fig.5.6b. In this architecture, both the discriminator and the generator circuits are built in two parts : an encoder ($E_{\boldsymbol{\theta}}$ or $E'_{\boldsymbol{\theta}}$) that takes the contextual data a and creates a quantum state that is then processed by another quantum variational circuit ($F_{\boldsymbol{\theta}}$ or $F'_{\boldsymbol{\theta}}$).

To train the our cQGAN, we apply the basic routine described in Algorithm 1. Similar to training a classic GAN, this training routine is decomposed into three phases. For a certain number of epochs e (line 1), we train successively the discriminator on real data (lines 1-1), the generator on the discriminator (lines 1-1), and the discriminator on the generator (lines 1-1). Each of these steps is repeated a certain number of times designated by the respective training counts $C_{D,X}, C_{D,G}, C_{G,D}$. These training counts are hyperparameters and require careful tuning to strike a suitable balance between the convergence of the discriminator and the generator. To train one of the agents we sample a subset of couples (a, b) of contextual data and target data (lines $8, 13, 18$). For each couple, we feed the agents with the contextual data and compute the derivatives of the appropriate loss function with respect to the para-

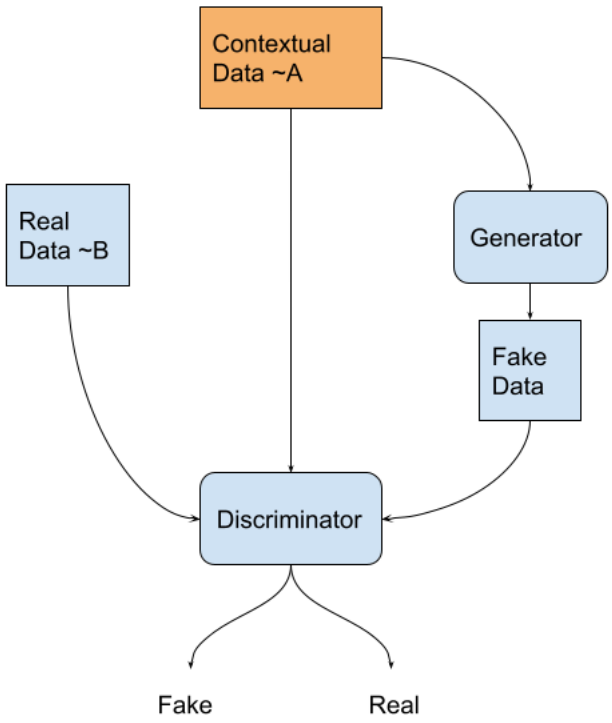
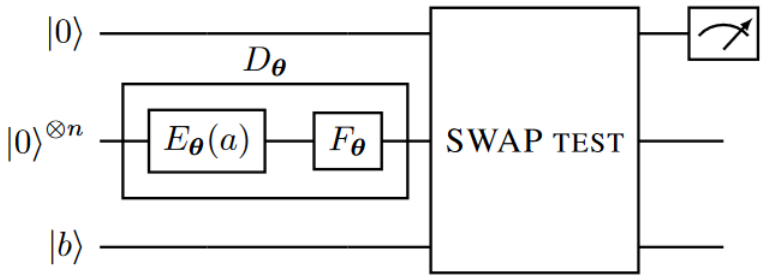
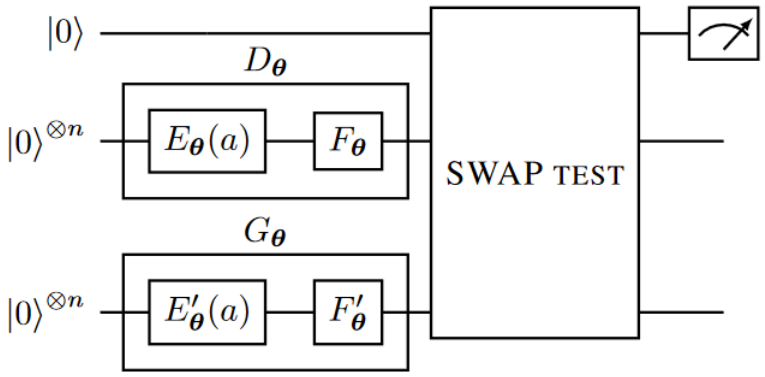


FIGURE 5.5 Pipeline of a conditional GAN.



(a) The discriminator and the real target data $|b\rangle$.



(b) The discriminator and the generator.

FIGURE 5.6 State fidelity conditional QGAN circuit design.

meter vector $\boldsymbol{\theta}$ (lines 1,1,1). Then we update $\boldsymbol{\theta}$ by taking a gradient step with an amplitude controlled by the learning rate α (lines 1,1,1).

5.3.2 SuDaI : Encoding through Successive Data Injection

With this architecture still comes the need to encode data larger than the number of qubits. To do so we introduce a new scheme called successive data injection. The principle is to incorporate data into the quantum state in order to, step by step, build a representation that encodes all the useful data for the task. This results in an encoding method that can accommodate a greater number of data points per number of available qubits, compared to what can be achieved using a basic angle encoding approach. The design consists of a stack of input layers which depend on the data and trainable parameters and variational layers which depend on trainable parameters only (see Fig. 5.7). This approach is inspired

Initialize :
Learning rate α
Circuit vector parameter $\boldsymbol{\theta}$
Dataset $X = (A, B)$
Number of epochs e
Training counts $C_{D,X}, C_{D,G}, C_{G,D}$
for $\xi \leftarrow 1$ **to** e **do**
`/* Training Discriminator on Real Data:` D, X `*/`
Sample $X_{D,X}$ *from* X *with* $|X_{D,X}| = C_{D,X}$
for $x = (a, b) \in X_{D,X}$ **do**
$\nabla_{\boldsymbol{\theta}} = \frac{dL_{D,X}(a,b)}{d\boldsymbol{\theta}}$
$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}}$
end
`/* Training Generator on Discriminator:` D, G `*/`
Sample $X_{D,G}$ *from* X *with* $|X_{D,G}| = C_{D,G}$
for $x = (a, b) \in X_{D,G}$ **do**
$\nabla_{\boldsymbol{\theta}} = \frac{dL_{D,G}(a)}{d\boldsymbol{\theta}}$
$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}}$
end
`/* Training Discriminator on Generator:` G, D `*/`
Sample $X_{G,D}$ *from* X *with* $|X_{G,D}| = C_{G,D}$
for $x = (a, b) \in X_{G,D}$ **do**
$\nabla_{\boldsymbol{\theta}} = \frac{dL_{G,D}(a)}{d\boldsymbol{\theta}}$
$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}}$
end
end

Algorithm 1: Training of the conditional quantum GAN

by previous works on quantum data reloading [68] and most notably [69] where the authors propose to re-upload data points at different points of the quantum variational circuit to improve performance. In this work we take it a step further by applying this to sequential data and uploading different data points at each layer.

This scheme consists of a succession of input layers $I_{\boldsymbol{\theta}}^{(k)}(a_k)$ and variational layers $V_{\boldsymbol{\theta}}^{(k)}$ for K data points $(a_1, ..., a_K)$ in a. The k^{th} input layer consists of a circuit that takes into account both the data point a_k and the parameter vector $\boldsymbol{\theta}$. The role of this layer is to inject data into the quantum state in the most general manner possible. possible The variational layer only depends on $\boldsymbol{\theta}$ and aims at preparing the quantum state for the next injection.

In theory, the circuit design of a such layer can take various forms. Numerous designs have

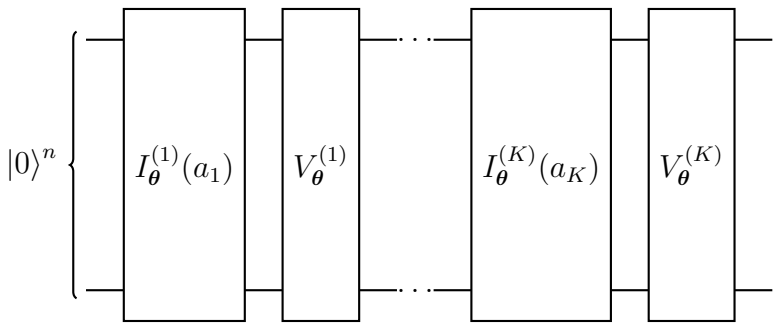


FIGURE 5.7 Architecture of the successive data injection circuit with K input and encodinglayers.

been proposed and studied in the literature, particularly in [57]. In this work, the authors benchmark different variational circuit designs by using two metrics. The first one is the Kullback-Leibler divergence for expressibility of a circuit, i.e. the ability of the circuit to explore most of the Hilbert Space. The second metric is the entangling capability which measures the ability of a circuit to generate entangled states. The design of the variational layer $V_{\boldsymbol{\theta}}^{(k)}$ used in this work is shown in Fig. 5.8 for 4 qubits. This design has been chosen among several ones described in [57], because of the good performance it exhibits in both metrics when used repeatedly while requiring a relatively low gate count per layer.

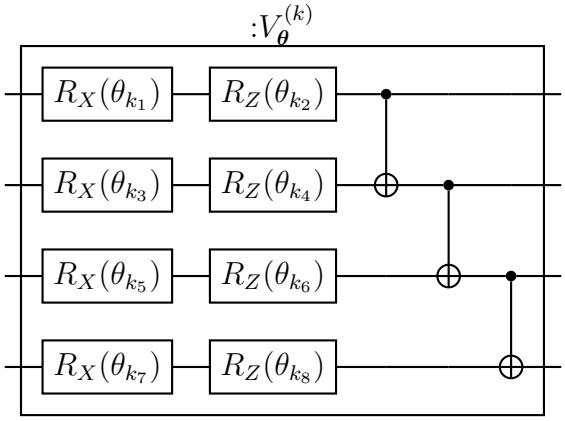


FIGURE 5.8 Variational circuit design.

The input layer specific design is described in Fig. 5.9 for four qubits. For each qubit, two parameters are used to specify the rotation angle of the R_Y gate : one to load the data and the other one as a weight that is learned during the training process. More precisely, the same data point a_k is encoded on each qubit but with different variational parameters. The repetition of such encoding has shown interesting result to increase the functionality of

quantum machine learning applications [68]. This creates a very general encoding procedure as all the qubits can be subject to rotations of different intensities.

It is also worth noting that when combining the input layer with the variational layer, three consecutive rotations R_Y, R_X and R_Z are applied on each qubit. These three rotations spanning the three axis are sufficient for expressing any one qubit unitary transformation [55]. An example of the complete circuit can be seen in Fig.5.10.

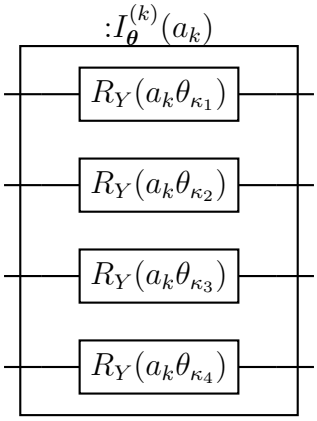


FIGURE 5.9 Design of the input layer sub-circuit.

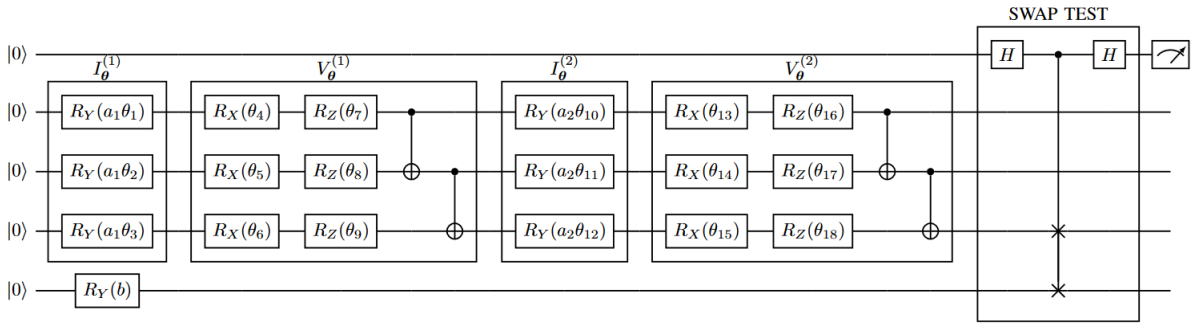


FIGURE 5.10 Example of a full circuit discriminator with two contextual data points, two qubits for the discriminator circuit and one qubit for the real data representation, the swap test is performed between the fourth and fifth qubits.

5.4 Anomaly Detection

5.4.1 DataSet

To apply our cQGAN to anomaly detection for time series we have used the Numenta Anomaly Benchmark [59] (NAB) database. Specifically, we chose real world data coming from

the sub-dataset called AWSCloudWatch. The time series consists of key performance indicators (KPI) recorded on servers during operation such as CPU usage. Anomalies in the server environment are then translated to anomalies in the time series of records of the different KPIs. The time series are univariate and the time step at which anomalies happen are labeled. The NAB also provides time windows around the anomalies that correspond to the time intervals inside which anomalies flagged by a model should be considered as true positives. A good anomaly detection system should therefore detect anomalies in those windows. One example of a time series with an anomaly and its associated time window can be seen in Fig. 5.11. The time windows are the areas in pale-red and the vertical red lines correspond to the timestamp with an anomalous data point.

An important constraint of NAB-AWSCLoudWatch is that certain time series have normal seasonal behavior with a characteristic time window of hundreds of time-steps as can be seen in Fig. 5.12. If we required our model to take them into account the SuDai encoding circuit would therefore necessitate hundreds of layers, a depth that goes way beyond what our simulation hardware can handle. Indeed the time complexity of training of a QVC of depth k grows quadraticaly in k : one run of the circuit is linear in terms of k but the number of runs to compute the gradient is proportional to the number of gates and consequently proportional to k. Thus, we chose a subset of the dataset that contains all the time series where periodic behavior is smaller than the time window chosen for our implementation, which is 10 time-steps.

This NAB-AWSCloudWatch dataset contains multiple time-series with different distributions, each is composed of thousands of time steps. To account for the specificities of the dataset we use a simple anomaly detection system. We implemented the proposed cQGAN using a continual learning strategy. That means the model is trained continually on a data stream.

5.4.2 Continual Learning on AWSCloudWatch Data

First, we normalize the data so that each point is contained in the interval $[0, 1]$. This can be accomplished without prior knowledge of the extremous data points in the series using the information available about the monitored KPI. For example, CPU usage, if measured in watts, is bounded by the maximum hardware power usage. This is the only preprocessing performed to the series.

A pseudocode description of the anomaly detection system using continual learning with

the proposed cQGAN is presented in Algorithm 2. At each time step (line 2), we use the cQGAN to generate a data point given the contextual data in a fixed time window (line 2) and compare it to the real data point. If the distance between the two is within a certain threshold η the data point is considered normal, if not it is flagged as an anomaly (lines 2-2). Then we train the model on the new data point by taking N gradient steps. The number of steps depends on the quantum loss (lines 2-2), which is obtained with the fidelity measurement between the generated and real data. Training is performed around anomalous points to enable the model to adapt to a potential new normal distribution that might differ from the pre-anomaly distribution. To accelerate the continual learning process, N can be set to 0 if the loss function is really close to its minimal value (smaller than some value ϵ). This formalizes the idea that if the system learned the conditional distribution and encounters a new data point that falls in this distribution, it does not need to learn further and can move on to the next time-step.

Initialize :
Time window Δ_t
Anomaly threshold η
Gradient step policy N
Circuit parameter vector $\boldsymbol{\theta}$
Learning rate α
Input : *Time series* $\{X_t\}_{t \leq T}$
Output : *Binary mapping* $\sigma : \{1, ..., T\} \rightarrow \{0, 1\}$
for $t \leftarrow 1$ **to** T **do**
  `/* Generate data and compare to detect an anomaly */`
  Generate $x'_t \leftarrow G_{\boldsymbol{\theta}}(X_{t-1}, ..., X_{t-\Delta_t})$
  if $|x_t - x'_t| < \eta$ **then**
    $\sigma(t) \leftarrow 0$
  else
    $\sigma(t) \leftarrow 1$
  end
  `/* Train the cQGAN on the new data point */`
  for $i \leftarrow 1$ **to** N **do**
    `Train cQGAN` (
      $a = (X_{t-1}, ..., X_{t-\Delta_t})$,
      $b = X_t$,
      $\boldsymbol{\theta}$,
      α
)
  end
end

Algorithm 2: Continual learning based anomaly detection system

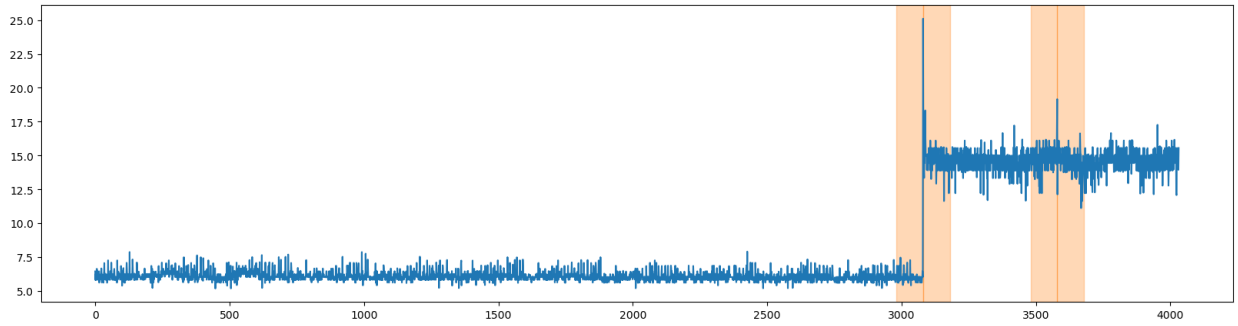


FIGURE 5.11 An example of a NAB time series monitoring CPU usage on a radio data system with two anomalies among 4000 time-steps. The anomalies are marked by the red lines while their corresponding time-windows for correct flagging are represented by the pale red areas.

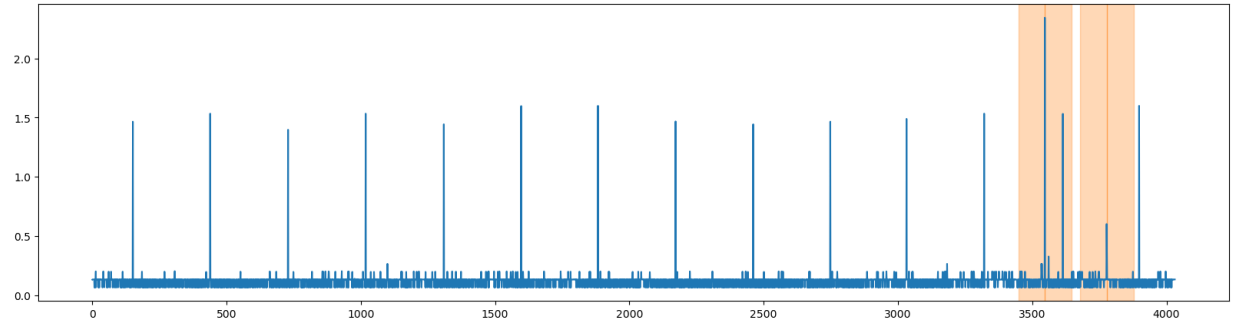


FIGURE 5.12 An example of a NAB time series with a seasonal behavior spanning over hundreds of timesteps.

5.4.3 Implementation Details

The implementation is accomplished with the Qiskit Python library [70] and its AerSimulator backend. Regarding the hyper-parameters, the learning rate is set to $\alpha = 0.005$. The cQGAN considers a time window of 10 timesteps to learn and make predictions. Lastly, the number of gradient steps per data point is $N = \lfloor 10 L_{D,X} \rfloor$ unless the quantum loss is smaller than a threshold $\epsilon = 0.05$ in which case $N = 0$. In our experiments on a AMD ryzen 5 3600 cpu, the execution of the model on a single time series is typically in the order of hours.

5.5 Results Analysis

5.5.1 Raw Results

Once the algorithm has run on the entire time series we can extract the time series of predictions and the time series of quantum loss scores. For the example illustrated in Fig.(5.11), we can visualize the result in Fig.(5.13). As we can see, the system learns to replicate the signal quite closely. When an anomaly occurs in the signal, a peak shows in the quantum loss at the anomaly point. Then the model adapts to the new "normal" signal and carries on. By examining the loss series, a point is flagged as an anomaly if its associated quantum loss is higher than the threshold η.

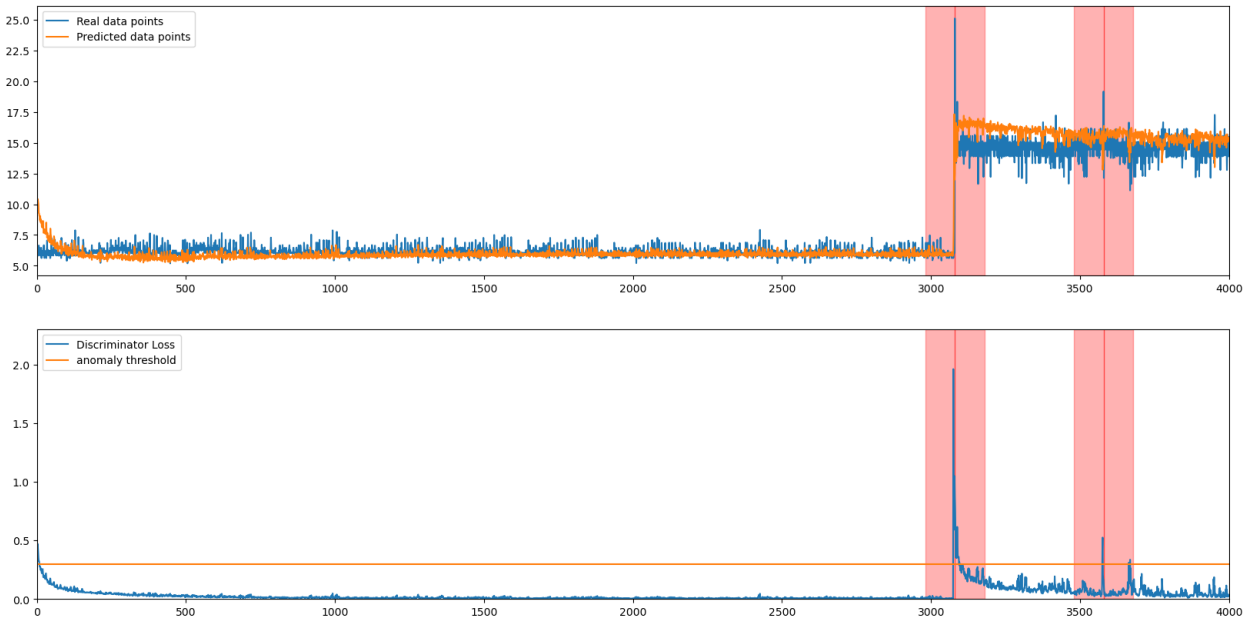


FIGURE 5.13 Example result after running the continual learning cQGAN on a time series with $\eta = 0.3$.

A noteworthy observation in the results is that either at the initial learning phase or when an anomaly occurs, the system takes a certain time to learn enough to reduce the loss below η. This results in many subsequent points wrongly flagged as anomalies. To avoid this pitfall one strategy is to prune some of these points by waiting for the system to adapt to the post anomaly distribution. This can be done either by waiting for a fixed number of time steps or by waiting until the loss gets smaller than a threshold that can be η or another value. For our case we used a waiting time of 30 timesteps

For the example of Fig.(5.13) and a value $\eta = 0.3$ the system detects both anomalies correctly as all values of the quantum loss above the threshold correspond to time-steps in the anomaly time windows of the time series. Two of them are the actual anomalies and the third one is an artefact of the second anomaly still contained inside the corresponding time window and is thus pruned by the NAB suggested methodology [59].

5.5.2 Dynamic Threshold

Selecting an appropriate value for η is crucial as excessively low values for η will lead to an excessive number of false positives while overly high values will miss many true anomalies and will generate a large amount of false negatives. However it would not be appropriate to manually chose a value for η that suits each time series, as this would essentially be equivalent to manually identifying anomalies. The scheme for choosing η should be consistent across all time series. Ideally we would chose a low value for η while avoiding to yield too many false positives. This ideal scenario is achieved when the loss reaches very low values because the model makes accurate predictions. Achieving this ideal scenario is however nearly impossible when dealing with noisy signals in one-point forecasting. Indeed, even if the model has learned a noisy data distribution and produces a noisy signal, the chance that the generated data points match the real ones is very small. So for noisy signals, the loss series will also be noisy and contain spikes as the model prediction may be quite different from reality from time to time. It is a challenge to prevent from flagging those spikes as anomalies as they are artefacts of the proposed method that is vulnerable to noise.

To address this challenge, we propose to use a dynamic threshold $\eta(t)$ that varies at each time step t. The threshold is computed as a linear model of an estimate of the noise in the loss series (Eq. (5.11)). The idea behind is that the necessary discrepancy with respect to normality should be proportional to the noise level for identifying spikes as anomalies. Therefore, we aim to have a high threshold when the signal is noisy and a low threshold when the signal has low noise. To establish this dynamic threshold, we proceed as follows.

Firstly, we characterize the noise at each given time. This primary need is performed in two steps. We start by defining the local variations around a time-step t as $Var(t, \delta)$ (see Eq.(5.9)), where δ is the temporal distance to the furthest neighbour considered. The value of δ is typically very small. In our implementation we use $\delta = 2$ time-steps. On the other hand, the local variation estimate considers only the largest variations. The rationale behind

this is that at such small timescales, prominent variations come from two sources : anomalies and noise. Indeed, a spike in the loss signifies that the model is surprised by the real data. If the model has already learned an underlying distribution, large surprises arise from elements outside of the distribution. These elements are then either noise or more significant changes in the distribution which can likely be considered as anomalies.

Secondly, we calculate a rough estimate of the dynamic noise level that allow excluding all the anomalies (and also some noise unfortunately). This is done through quantile calculation. We assume that anomalies are rare representing less than 0.5% of the data points, and they should be located at the far end of the noise distribution. Therefore, we fix the quantile at $p = 0.95$. This means we assume that 95% of lower noises computed are considered as true noises. In our implementation, the noise estimator is applied within a large time window with $\tau = 500$, i.e. within 500 timestamps before the current time. The Eq. (5.10) allows calculating the estimate of the noise level.

Finally, we construct the dynamic threshold η (see Eq. (5.11)) by starting with a base threshold b. Then we add the noise level estimation weighted by a parameter a. Ideally b should be as small as possible while taking into account the ϵ parameter used earlier as ϵ represents the acceptable error for the model. We chose $b = 6 \times \epsilon = 0.3$ to be robust to small variations especially when they are close to no noise overall. The key is to select a value for a that scales the noise estimation to a useful level when compared to the loss. We experimentally pick $a = 2$.

$$Var(t, \delta) = max\{|L_t - L_i|\}_{|i-t| \leq \delta} \tag{5.9}$$

$$Noise(t, \tau) = Quantile(p, \{Var(s, \delta)\}_{t-\tau<s<t}) \tag{5.10}$$

$$\eta(t) = a\, Noise(t, \tau) + b \tag{5.11}$$

5.5.3 Experimental Results

The results concerning the 8 time series on which we tested our model are provided in Table 5.1. As suggested by the NAB methodology we consider all positives within one anomaly time window to be the same and reduced to the earliest detection. We did not include the numbers of True Negatives in this table as they are not very informative in this context. Most data points are not anomalies and are not flagged. In other words, almost all the points outside of anomaly time windows are true negatives.

Time series name	Number of anomalies	Number of positives	True positives	False positives	False negatives
ec2_cpu_utilization_5f5533	2	2	2	0	0
ec2_cpu_utilization_825cc2	2	3	2	1	0
ec2_cpu_utilization_ac20cd	1	2	1	1	0
ec2_network_in_257a54	1	1	1	0	0
elb_request_count_8c0756	2	3	2	1	0
iio_us-east-1_i-a2eb1cd9_NetworkIn	2	1	1	0	1
rds_cpu_utilization_cc0c53	2	2	2	0	0
rds_cpu_utilization_e47b3b	2	2	2	0	0
Total	14	16	13	3	1

TABLEAU 5.1 Results

From the Table 5.1, we can see there is only one false negative across all the data. It corresponds to a small anomalous spike in "iio_us-east-1_i-a2eb1cd9_NetworkIn". There are a few false positives, two of them corresponding to normal (as per the labels given in NAB) but quite drastic changes in the distribution. These instances occur in datastreams "ec2_cpu_utilization_825cc2" and "ec2_cpu_utilization_ac20cd". In fact, both of those false positives are due to a shift back to a previous normal distribution. Since our model works on $\Delta_t = 10$ time steps, it has no chance of remembering the old distribution. Therefore the model considers this change as an anomaly. The last false positive is due to noise in "elb_request_count_8c0756", the most noisy stream used. The dynamic threshold allows to avoid numerous spikes caused by noise. Still, one of these spikes got flagged as an anomaly by the system as can be seen in Fig. 5.15 of the Appendix. Moreover by looking at the curves for such a noisy signal, the dynamic threshold is operating at its limit with several noise spikes approaching the threshold closely. The rest of the results can be seen in the appendix in Fig.5.14 and Fig.5.15.

5.6 Conclusion and Future Works

In this paper we introduced a novel method for encoding high dimensional data using a smaller number of qubits compared to the most commonly used method which is angle encoding. This method encodes high dimensional data into the quantum state while still using a linear number of gates. This allows us to process more data than with the usual angle encoding method since the number of qubits necessary needs not to scale with the number of data

points at hand. The number of qubits does not depend on the size of the data but rather on the functionnality that needs to be expressed. This method also retains a complexity advantage over techniques such as amplitude encoding. In short, SuDaI solves the problem of the dimensionality of the data being encoded by shifting the limit from number of qubits to the depth of the circuit. In general, increasing the depth of a quantum circuit generates more noise which is detrimental to the performance of an algorithm. However, quantum variational algorithms have been proven to be more robust to noise both theoretically and experimentally [71].

Besides, we have applied the new encoding method to build a cQGAN for anomaly detection. For this application, we have devised a dynamic threshold that incorporates the noise level in the loss signal to enhance the precision of anomaly detection, enabling the model to handle noisy signals better. While the suggested dynamic threshold shows promising results, the challenge of noise has not been entirely eliminated. One potential solution to address this issue would be transitioning from a forecasting-based approach to a distribution-based approach, which would account for noise without requiring post-processing.

It is important to highlight that the experiments were conducted using network data but the model can be readily applied to anomaly detection in any kind of sequential data, provided that the time window hypothesis stands. Typical use cases might include financial time series, seismic or meteorologic data among others. Furthermore, SuDaI could be generalized for encoding generic data into a quantum state in non-conditional quantum GANs. This would enable the utilization of higher dimensional data in quantum variational applications, even within the constraints of the current limitations on the number of available qubits.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada, Prompt Innov, Thales and Zetane Systems, for the financial support of this research.

Appendix

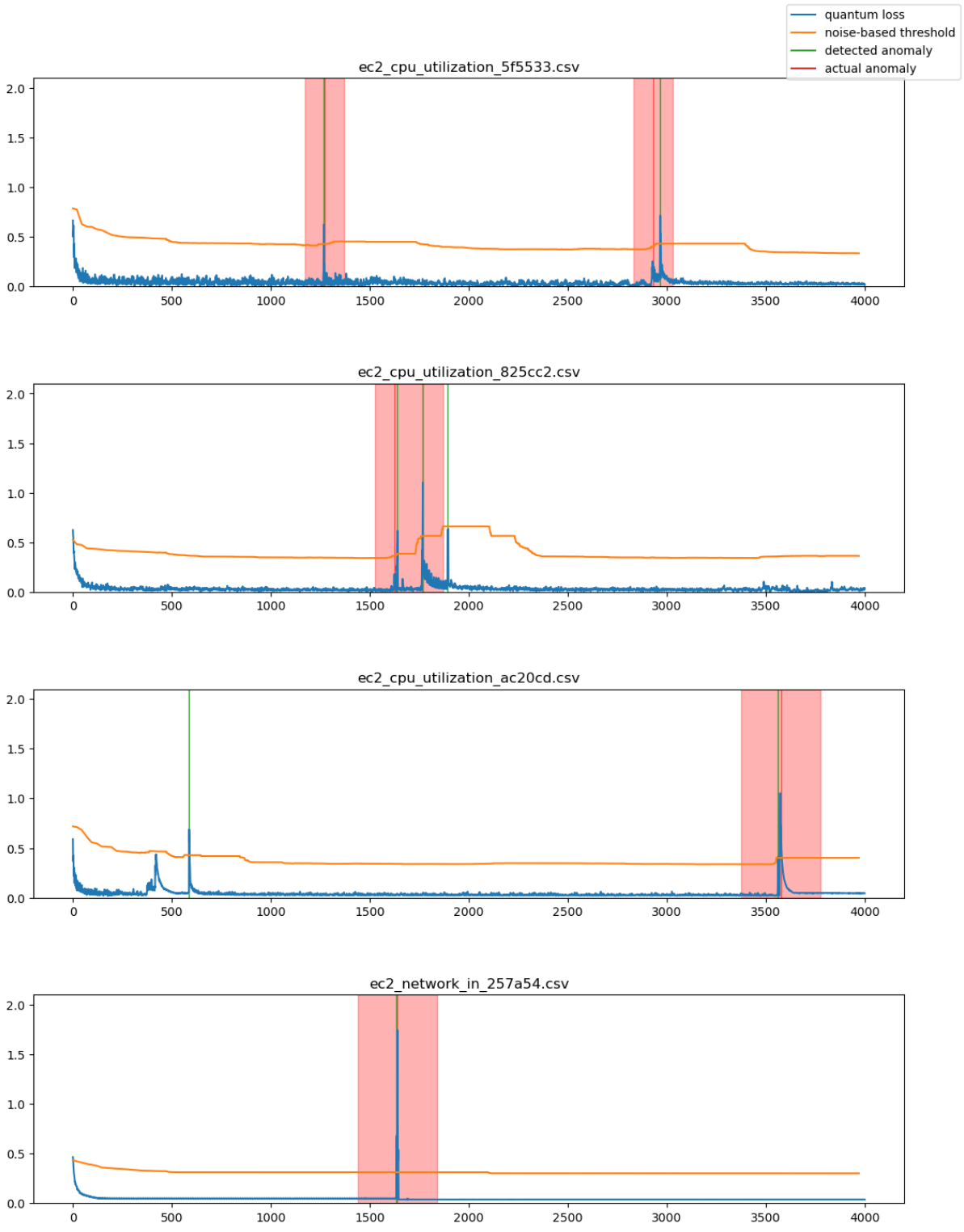


FIGURE 5.14 Full results for all selected time series part 1

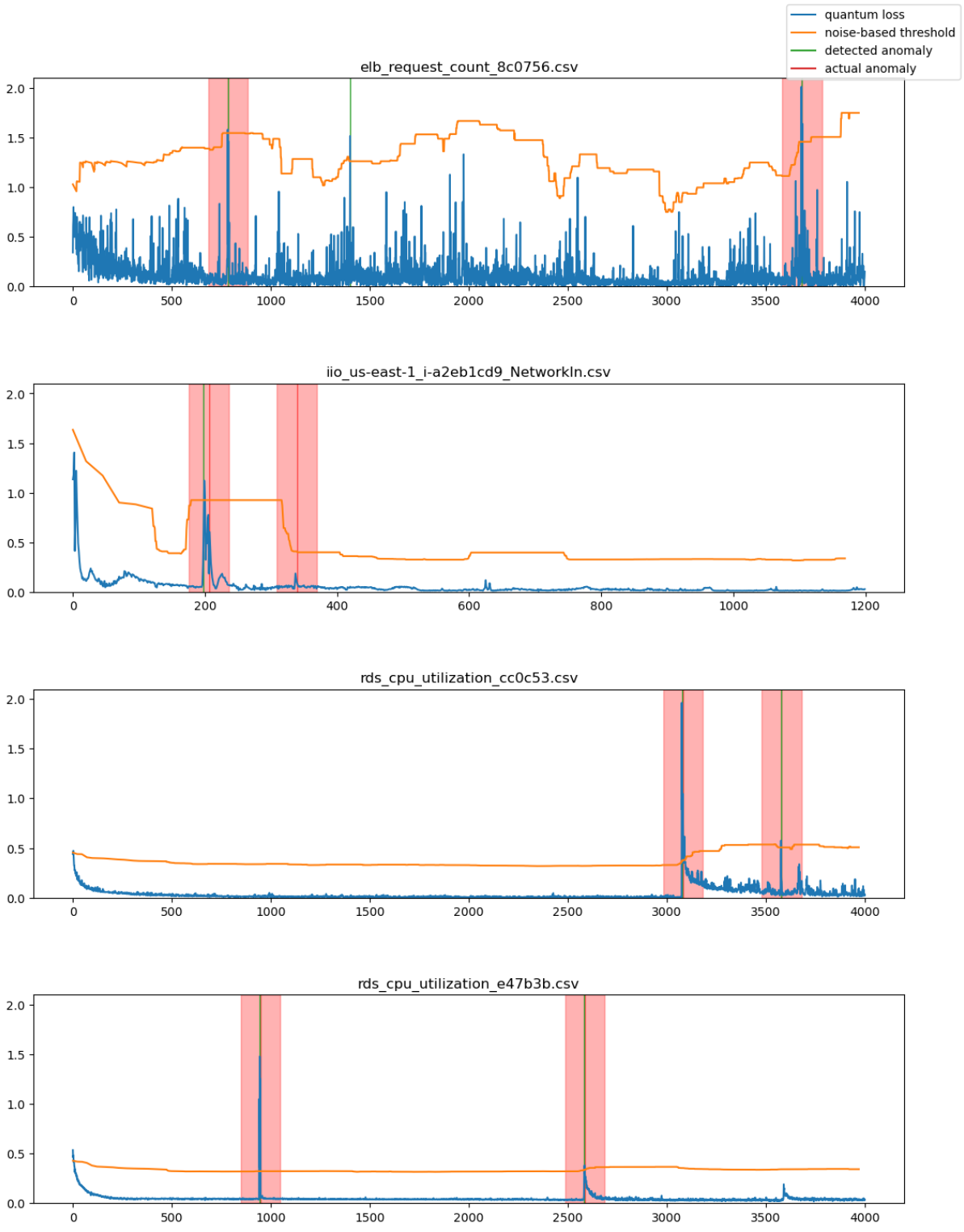


FIGURE 5.15 Full results for all selected time series part 2

CHAPITRE 6 DISCUSSION SUR LES CHOIX ET LES TRAVAUX RÉALISÉS

Ce travail de recherche est la première étape d'un plus large projet de recherche sur l'utilisation de QGANs pour la détection d'anomalies pour les réseaux. Son objectif principal était la mise au point et la réalisation d'un système de détection d'anomalies pour les séries temporelles utilisant un GAN quantique. Cet objectif a été atteint en développant un système complet qui utilise une nouvelle pour l'injection des données dans l'état quantique et qui incorpore des méthodes de l'état de l'art. La méthode développée à permis d'obtenir des résultats prometteurs sur un jeu de donnée commun pour le test de ce type de systèmes. Cependant, de nombreuses difficultés et limites ont été rencontrées lors de la réalisation de ce travail, mais aussi à la lumière des résultats obtenus. Ce chapitre vise les exposer tout en expliquant les choix et méthodes réalisées pour contourner ces difficultés, et ce, dans une démarche favorisant la réutilisation du travail effectué. Puisque ce travail est avant tout un travail d'exploration et est le premier d'une suite d'autres dans le cadre d'un large projet, il convient de rendre compte d'un maximum de connaissances théoriques et pratiques obtenues pour permettre la progression des recherches futures dans cette voie.

6.1 Données

Comme expliqué plus tout au long de ce mémoire, la question des données est centrale au projet de recherche. Le type de données utilisé, les séries temporelles, est la raison de bon nombre de complications et de difficultés, tant au niveau des données elles-mêmes qu'à leur incorporation dans un projet impliquant de l'informatique quantique. Premièrement, comme suggéré en chapitre 3, les données réseau typiquement utilisées pour la détection d'anomalies ne sont pas directement adaptées à l'application aux méthodes basées sur les séries temporelles. Des transformations doivent être réalisées pour permettre leur exploitation en tant que séries temporelles. Les jeux de données sont typiquement composés d'événements caractérisés par de nombreux attributs. Crucialement, bon nombre de ces jeux de données ne sont pas horodatés et ne sont donc pas convertissables en séries temporelles. Pour ceux qui sont horodatés comme par exemple le Coburg Intrusion Detection Dataset, il reste la question de transformer toutes les caractéristiques en composantes propices à la détection des anomalies réseau que l'on veut détecter.

Une approche simple et réalisée au cours de la maitrise pour le jeu de donnée CIDDS est la réalisation d'un prétraitement capable d'extraire certaines données numériques comme le

nombre de paquets échangé ou le nombre d'octets échangés ou le nombre d'évenements par unité de temps. L'avantage de cette approche est sa généralisation à d'autres jeux de données horodatés : ces caractéristiques d'événements se retrouvent au travers de tous les jeux de données réseau dont nous avons eu connaissance. Seulement, ce type de caractéristiques est adapté uniquement à la détection de certaines anomalies : celles qui induisent une augmentation de l'activité dans le réseau lui-même. On peut par exemple citer parmi elles les attaques DDoS. D'autres anomalies correspondant par exemple a des attaques par scan ne correspondent pas forcément à une variation de certaines caractéristiques comme par exemple le nombre de paquets échangés. On aura donc du mal à les détecter à partir de ces caractéristiques.

Une approche plus simple et tout de même assez généraliste est de s'intéresser aux séries temporelles surveillant des indicateurs de performance clés. En effet, bon nombre d'anomalies liées à des attaques sur des réseaux se retranscrivent sur les machines et sous-composants de ce réseau et on peut alors observer dysfonctionnement de leur comportement. L'idée est donc d'enregistrer des signaux qui permettent de rendre compte de l'activité de ces composants. Un signal typique est l'utilisation d'un processeur de machines du réseau ou l'utilisation de la mémoire vive. C'est donc ce type de donnée qui a été préféré pour l'application de la proposition de QGAN.

AU vu des limitations liées à l'approche quantique et comme ce travail est le premier du projet, on s'est restreint aux séries temporelles univariées. Dans un aspect pratique, ce choix rend plus facile l'élaboration du système puisque les données sont plus simples mais complique le choix de la série temporelle utilisée. Il faut réduire la complexité du système à un seul signal, ce qui représente une perte d'information pour un système complexe.

6.2 Choix relatifs au QGAN

Comme expliqué dans la revue de littérature, deux grands types de QGANs ont été proposés dans la littérature, ceux basés sur la mesure de la fidélité de l'état quantique et les autres qui remplacent simplement les agents classiques d'un GAN par des agents quantiques. Au delà, des arguments théoriques déjà expliqués sur les différentes méthodes d'optimisation de la fonction quantique, les méthodes hybrides rencontrent des difficultés quant à la convergence équilibrée entre le discriminateur et le générateur. En effet, le problème de la convergence d'un GAN classique est déjà bien connu : par exemple on observe qu'un des agents, typiquement le discriminateur, devient bien trop bon par rapport à l'autre qui ne peut plus tenir la cadence d'apprentissage. Au vu des expérimentations réalisés avec ce genre de méthode,

on rencontre ces problèmes d'équilibre. Cependant, avec les approches totalement quantiques retenues pour la proposition, ce problème est bien moindre. La réponse qui semble la plus probable est que la symétrie de design entre le discriminateur et le générateur dans les QGANs à fidélité d'état se répercute en un équilibre de vitesse d'apprentissage bien meilleur entre les deux agents. Cette hypothèse est corroborée par le fait qu'expérimentalement les méthodes les plus susceptibles à ce déséquilibre étaient celles incorporant des structures classiques d'échelles différentes des structures quantiques, comme par exemple avec un discriminateur classique puissant et un générateur quantique. Ces raisons expliquent le choix du type de QGAN utilisé pour la réalisation de ce travail.

6.3 Implémentation et optimisations

La vitesse d'exécution du code a été un point clé pour l'expérimentation puisque les méthodes d'apprentissage machine, même quantique, nécessitent de longs temps pour obtenir une convergence des modèles. Être capable d'itérer rapidement est également important dans l'exploration de nouvelles voies et prototypes de modèles. Un effort a été réalisé dans ce sens puisque nous avons utilisé les conduites les plus récentes et recommandées dans la librairie qiskit. Un travail d'optimisation de certaines tâches réalisées par des fonctions dédiée de la librairie a même été fait en les remplaçant par des structures issues de numpy pour accélérer encore le processus d'expérimentation. En termes de matériel, le simulateur classique n'a pas été un facteur limitant en termes de puissance de calcul sur un processeur 4.2 GHz commun avec une utilisation en dessous de 5%. Un entrainement sur un pas de temps pour un circuit quantique tel que celui utilisé pour obtenir les résultats de l'article prenait un temps de l'ordre de grandeur de la minute. Ce qui multiplié par les milliers de pas de temps contenus dans les séries temporelles commence à représenter des temps comparables à des journées. On utilise toutefois des optimisations dans la stratégie d'apprentissage pour ne s'entrainer que lors ce que c'est nécessaire. Effectivement, plus le modèle effectue de bonnes prédictions, plus on diminue l'entrainement puisqu'une fois la distribution apprise, il n'y a plus besoin de continuer à dépenser des ressources pour continuer l'apprentissage. Il faut toutefois nuancer ces résultats puisque les tailles de circuits ont été poussés à leur maximum dans les limites des contraintes possibles (10 qubits pour le générateur et le discriminateur). Avec des circuits plus petits traitant moins de données que celui proposé tels que ceux présents dans la littérature, les temps d'entrainement sont acceptables [15–17].

6.4 Analyse des résultats et limites

Les résultats obtenus dans l'article sont prometteurs d'un point de vue de détection d'anomalies, cependant les points d'échecs sur les données offrent des perspectives d'amélioration pour la proposition. Premièrement, malgré l'introduction d'un système correctif partiel, l'approche par prédiction est assez sensible au bruit dans le signal réel. Si il y a beaucoup de bruits dans le signal, la distribution apprise par le QGAN sera assez dispersée et donc il y a de bonnes chances que même si la distribution est la bonne le point prédit soit éloigné de la réalité. D'autre part, le concept de la proposition est de fabriquer un système qui a une mémoire de taille fixée (ou fenêtre de temps contextuelle). Donc pour prendre en compte des phénomènes qui occurrent sur de longues périodes, il faut au moins augmenter la taille de cette mémoire pour qu'elle soit aussi longue que ces durées liées aux phénomènes. Cependant, comme expliqué dans la section précédente, les résultats obtenus sont déjà à la limite de l'implémentation en termes de faisabilité expérimentale avec le matériel disponible actuellement. Comme le temps d'apprentissage dépend quadratiquement de la longueur de cette fenêtre contextuelle qui fait office de mémoire, on ne peut pas aller beaucoup plus loin avec l'implémentation actuelle.

CHAPITRE 7 CONCLUSION

Ce mémoire propose une nouvelle méthode d'injection de données dans un circuit quantique en tant que réponse au problème de la dimensionnalité des données issues de séries temporelles dans le cadre de l'apprentissage machine quantique. Cette méthode est incorporée dans un design de QGAN de l'état de l'art en la matière afin d'être la pièce centrale d'un système de détection d'anomalies pour les séries temporelles. Pour ce faire, une implémentation a été réalisée en python tout en cherchant à optimiser les processus disponibles pour réduire la durée des calculs. On obtient des résultats satisfaisants pour la détection d'anomalies pour les séries temporelles applicables à notre modèle.

À la lueur des discussions tenues dans le chapitre précédent, on peut proposer plusieurs pistes d'améliorations et de travaux futurs potentiels basés sur celui présenté dans ce mémoire :
Premièrement, il y a encore des optimisations à réaliser en termes de vitesse d'exécution du code, par exemple, on peut utiliser un simulateur quantique qui prend avantage de matériel dédié au calcul comme une carte graphique puisque simuler un état quantique et son évolution revient à multiplier des matrices. Une tache pour laquelle certaines optimisations matérielles existent déjà. On peut également penser à paralléliser différentes étapes du processus d'entrainement pour encore améliorer la vitesse d'exécution.
Deuxièmement, à la lueur de la réflexion sur l'aspect court-termiste de la mémoire du design de QGAN conditionnel proposé, il y a des améliorations à faire sur quelles données considérer parmi les données précédentes pour construire la donnée suivante. Dans le cadre classique, des mécanismes comme les Long Short Term Memory ou les mécanismes à base d'attention ont permis la prise en compte de données plus lointaines dans des données séquentielles, des approches avec le même objectif dans le cadre quantique seront sans doute bénéfiques.
Troisièmement, il pourrait être utile d'explorer le fait de considérer l'état quantique en sortie du modèle génératif comme la sortie à part entière du GAN. À la place d'interpréter les amplitudes des états de la base computationnelle comme des valeurs numériques (ce qui est l'approche privilégiée à la fois dans la littérature et dans ce travail), on peut interpréter ces amplitudes comme une distribution de probabilité. Dans le cadre de la détection d'anomalies pour les séries temporelles, si on s'entraine à générer la distribution pour le prochain point, il sera facile de comparer une donnée a une distribution prédite pour déterminer la vraisemblance d'appartenance à la classe des données normales.

RÉFÉRENCES

[1] S. Monteith, M. Bauer, M. Alda, J. Geddes, P. C. Whybrow et T. Glenn, "Increasing cybercrime since the pandemic : Concerns for psychiatry," *Current psychiatry reports*, vol. 23, p. 1–9, 2021.

[2] B. Cashell, W. D. Jackson, M. Jickling et B. Webel, "The economic impact of cyber-attacks," *Congressional research service documents, CRS RL32331 (Washington DC)*, vol. 2, 2004.

[3] J. J. McCarthy, *Climate change 2001 : impacts, adaptation, and vulnerability : contribution of Working Group II to the third assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2001, vol. 2.

[4] S. Doocy, A. Daniels, C. Packer, A. Dick et T. D. Kirsch, "The human impact of earthquakes : a historical review of events 1980-2009 and systematic literature review," *PLoS currents*, vol. 5, 2013.

[5] W. J. W. Botzen, O. Deschenes et M. Sanders, "The Economic Impacts of Natural Disasters : A Review of Models and Empirical Studies," *Review of Environmental Economics and Policy*, vol. 13, n°. 2, p. 167–188, juill. 2019, publisher : The University of Chicago Press. [En ligne]. Disponible : https://www.journals.uchicago.edu/doi/full/10.1093/reep/rez004

[6] J. M. Craddock, "The Analysis of Meteorological Time Series for Use in Forecasting," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 15, n°. 2, p. 167–190, 1965, publisher : [Royal Statistical Society, Wiley]. [En ligne]. Disponible : https://www.jstor.org/stable/2987390

[7] R. S. Tsay, "Time Series Model Specification in the Presence of Outliers," *Journal of the American Statistical Association*, vol. 81, n°. 393, p. 132–141, mars 1986, publisher : Taylor & Francis _eprint : https ://www.tandfonline.com/doi/pdf/10.1080/01621459.1986.10478250. [En ligne]. Disponible : https://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478250

[8] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa,

D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven et J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, nº. 7779, p. 505–510, oct. 2019, number : 7779 Publisher : Nature Publishing Group. [En ligne]. Disponible : https://www.nature.com/articles/s41586-019-1666-5

[9] G. Pang, C. Shen, L. Cao et A. V. D. Hengel, "Deep learning for anomaly detection : A review," *ACM Comput. Surv.*, vol. 54, nº. 2, mar 2021. [En ligne]. Disponible : https://doi.org/10.1145/3439950

[10] M. Braei et S. Wagner, "Anomaly Detection in Univariate Time-series : A Survey on the State-of-the-Art," avr. 2020. [En ligne]. Disponible : https://arxiv.org/abs/2004.00433v1

[11] S. Akcay, A. Atapour-Abarghouei et T. P. Breckon, "GANomaly : Semi-Supervised Anomaly Detection via Adversarial Training," mai 2018. [En ligne]. Disponible : https://arxiv.org/abs/1805.06725v3

[12] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs et U. Schmidt-Erfurth, "f-AnoGAN : Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, vol. 54, p. 30–44, mai 2019. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1361841518302640

[13] S. Lloyd et C. Weedbrook, "Quantum Generative Adversarial Learning," *Physical Review Letters*, vol. 121, nº. 4, p. 040502, juill. 2018, publisher : American Physical Society. [En ligne]. Disponible : https://link.aps.org/doi/10.1103/PhysRevLett.121.040502

[14] C. Zoufal, A. Lucchi et S. Woerner, "Quantum Generative Adversarial Networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, nº. 1, p. 1–9, nov. 2019, number : 1 Publisher : Nature Publishing Group. [En ligne]. Disponible : https://www.nature.com/articles/s41534-019-0223-2

[15] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi et H. Neven, "Entangling Quantum Generative Adversarial Networks," *Physical Review Letters*, vol. 128, nº. 22, p. 220505, juin 2022, publisher : American Physical Society. [En ligne]. Disponible : https://link.aps.org/doi/10.1103/PhysRevLett.128.220505

[16] C. Chu, G. Skipper, M. Swany et F. Chen, "IQGAN : Robust Quantum Generative Adversarial Network for Image Synthesis On NISQ Devices," oct. 2022, arXiv :2210.16857 [quant-ph]. [En ligne]. Disponible : http://arxiv.org/abs/2210.16857

[17] S. A. Stein, B. Baheri, D. Chen, Y. Mao, Q. Guan, A. Li, B. Fang et S. Xu, "QuGAN : A Quantum State Fidelity based Generative Adversarial Network," dans *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, oct. 2021, p. 71–81.

[18] P. C. Mahalanobis, "On the generalized distance in statistics," *Sankhyā : The Indian Journal of Statistics, Series A (2008-)*, vol. 80, p. pp. S1–S7, 2018. [En ligne]. Disponible : https://www.jstor.org/stable/48723335

[19] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich et K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, n°. 5, p. 756–795, 2021.

[20] C. Leys, O. Klein, Y. Dominicy et C. Ley, "Detecting multivariate outliers : Use a robust variant of the mahalanobis distance," *Journal of experimental social psychology*, vol. 74, p. 150–156, 2018.

[21] S. Roberts et L. Tarassenko, "A probabilistic resource allocating network for novelty detection," *Neural Computation*, vol. 6, n°. 2, p. 270–284, 1994.

[22] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, n°. 3, p. 1065–1076, 1962.

[23] J. Kim et C. D. Scott, "Robust kernel density estimation," *The Journal of Machine Learning Research*, vol. 13, n°. 1, p. 2529–2565, 2012.

[24] J. Andrews, T. Tanay, E. J. Morton et L. D. Griffin, "Transfer representation-learning for anomaly detection." JMLR, 2016.

[25] D. Xu, E. Ricci, Y. Yan, J. Song et N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," 2015.

[26] D. Achlioptas, "Database-friendly random projections : Johnson-lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, n°. 4, p. 671–687, 2003.

[27] P. Li, T. J. Hastie et K. W. Church, "Very sparse random projections," dans *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, p. 287–296.

[28] S. Wold, K. Esbensen et P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, n°. 1-3, p. 37–52, 1987.

[29] E. J. Candès, X. Li, Y. Ma et J. Wright, "Robust principal component analysis ?" *Journal of the ACM (JACM)*, vol. 58, n°. 3, p. 1–37, 2011.

[30] R. Tudor Ionescu, S. Smeureanu, B. Alexe et M. Popescu, "Unmasking the abnormal events in video," dans *Proceedings of the IEEE international conference on computer vision*, 2017, p. 2895–2903.

[31] G. E. Hinton et R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, n°. 5786, p. 504–507, 2006.

[32] Y. Wang, H. Yao et S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, p. 232–242, 2016.

[33] S. Hawkins, H. He, G. Williams et R. Baxter, "Outlier detection using replicator neural networks," dans *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2002, p. 170–180.

[34] J. Chen, S. Sathe, C. Aggarwal et D. Turaga, "Outlier detection with autoencoder ensembles," dans *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, p. 90–98.

[35] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville et Y. Bengio, "Generative Adversarial Networks," juin 2014. [En ligne]. Disponible : https://arxiv.org/abs/1406.2661v1

[36] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth et G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," mars 2017. [En ligne]. Disponible : https://arxiv.org/abs/1703.05921v1

[37] H. Zenati, C. S. Foo, B. Lecouat, G. Manek et V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," mai 2019, arXiv :1802.06222 [cs, stat]. [En ligne]. Disponible : http://arxiv.org/abs/1802.06222

[38] L. Metz, B. Poole, D. Pfau et J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv :1611.02163*, 2016.

[39] A. Blázquez-García, A. Conde, U. Mori et J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys (CSUR)*, vol. 54, n°. 3, p. 1–33, 2021.

[40] G. B. Moody et R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE engineering in medicine and biology magazine*, vol. 20, n°. 3, p. 45–50, 2001.

[41] B. Zhou, S. Liu, B. Hooi, X. Cheng et J. Ye, "Beatgan : Anomalous rhythm detection using adversarially generated time series." dans *IJCAI*, vol. 2019, 2019, p. 4433–4439.

[42] Z. Li, W. Chen et D. Pei, "Robust and unsupervised kpi anomaly detection based on conditional variational autoencoder," dans *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2018, p. 1–9.

[43] M. Sakurada et T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," dans *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, p. 4–11.

[44] A. Siffer, P.-A. Fouque, A. Termier et C. Largouet, “Anomaly detection in streams with extreme value theory,” dans *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, p. 1067–1075.

[45] A. Aboode, “Anomaly detection in time series data based on holt-winters method,” 2018.

[46] P. Malhotra, L. Vig, G. Shroff, P. Agarwal *et al.*, “Long short term memory networks for anomaly detection in time series.” dans *Esann*, vol. 2015, 2015, p. 89.

[47] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong et Q. Zhang, “Multivariate time-series anomaly detection via graph attention network,” dans *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, p. 841–850.

[48] Y. Wang, L. Han, W. Liu, S. Yang et Y. Gao, “Study on wavelet neural network based anomaly detection in ocean observing data series,” *Ocean Engineering*, vol. 186, p. 106129, 2019.

[49] K. E. Smith et A. O. Smith, “Conditional gan for timeseries generation,” *arXiv preprint arXiv :2006.16477*, 2020.

[50] M. Schuld, I. Sinayskiy et F. Petruccione, “An introduction to quantum machine learning,” *Contemporary Physics*, vol. 56, n°. 2, p. 172–185, avr. 2015, publisher : Taylor & Francis _eprint : https ://doi.org/10.1080/00107514.2014.964942. [En ligne]. Disponible : https://doi.org/10.1080/00107514.2014.964942

[51] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe et S. Lloyd, “Quantum machine learning,” *Nature*, vol. 549, n°. 7671, p. 195–202, sept. 2017, number : 7671 Publisher : Nature Publishing Group. [En ligne]. Disponible : https://www.nature.com/articles/nature23474

[52] P.-L. Dallaire-Demers et N. Killoran, “Quantum generative adversarial networks,” *Physical Review A*, vol. 98, n°. 1, p. 012324, juill. 2018, place : College Pk Publisher : Amer Physical Soc WOS :000439409500007. [En ligne]. Disponible : https://journals.aps.org/pra/abstract/10.1103/PhysRevA.98.012324

[53] G. E. Crooks, “Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition,” mai 2019. [En ligne]. Disponible : https: //arxiv.org/abs/1905.13311v1

[54] D. Wierichs, J. Izaac, C. Wang et C. Y.-Y. Lin, “General parameter-shift rules for quantum gradients,” juill. 2021. [En ligne]. Disponible : https://arxiv.org/abs/2107.12390v3

[55] M. A. Nielsen et I. L. Chuang, “Quantum computation and quantum information,” *Phys. Today*, vol. 54, n°. 2, p. 60, 2001.

[56] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio et P. J. Coles, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, n°. 9, p. 625–644, sept. 2021, number : 9 Publisher : Nature Publishing Group. [En ligne]. Disponible : https://www.nature.com/articles/s42254-021-00348-9

[57] S. Sim, P. D. Johnson et A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, n°. 12, p. 1900070, déc. 2019, arXiv :1905.10876 [quant-ph]. [En ligne]. Disponible : http://arxiv.org/abs/1905.10876

[58] M. Tavallaee, E. Bagheri, W. Lu et A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," dans *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, p. 1–6.

[59] S. Ahmad, A. Lavin, S. Purdy et Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, p. 134–147, 2017.

[60] R. S. Tsay, *Analysis of Financial Time Series*. John Wiley & Sons, sept. 2005, google-Books-ID : ddL4tTLb_08C.

[61] Q. Wu et Z. Shao, "Network Anomaly Detection Using Time Series Analysis," dans *Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services - (icas-isns'05)*, oct. 2005, p. 42–42, iSSN : 2168-1872.

[62] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang et N. Ding, "Gan-based anomaly detection : A review," *Neurocomputing*, vol. 493, p. 497–535, 2022.

[63] M. Mirza et S. Osindero, "Conditional generative adversarial nets," 2014.

[64] N. P. de Leon, K. M. Itoh, D. Kim, K. K. Mehta, T. E. Northup, H. Paik, B. S. Palmer, N. Samarth, S. Sangtawesin et D. W. Steuerman, "Materials challenges and opportunities for quantum computing hardware," *Science*, vol. 372, n°. 6539, p. eabb2823, avr. 2021, publisher : American Association for the Advancement of Science. [En ligne]. Disponible : https://www.science.org/doi/full/10.1126/science.abb2823

[65] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, "Evidence for the utility of quantum computing before fault tolerance," *Nature*, vol. 618, n°. 7965, p. 500–505, 2023.

[66] A. Kutvonen, K. Fujii et T. Sagawa, "Optimizing a quantum reservoir computer for time series prediction," *Scientific reports*, vol. 10, n°. 1, p. 14687, 2020.

[67] Z. Mlika, S. Cherkaoui, J. F. Laprade et S. Corbeil-Letourneau, "User trajectory prediction in mobile wireless networks using quantum reservoir computing," *IET Quantum Communication*, 2023.

[68] M. Schuld, R. Sweke et J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Physical Review A*, vol. 103, n°. 3, p. 032430, 2021.

[69] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster et J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.

[70] Qiskit contributors, "Qiskit : An open-source framework for quantum computing," 2023.

[71] K. Sharma, S. Khatri, M. Cerezo et P. J. Coles, "Noise resilience of variational quantum compiling," *New Journal of Physics*, vol. 22, n°. 4, p. 043006, 2020.

ANNEXE A SÉRIES TEMPORELLES UTILISÉES

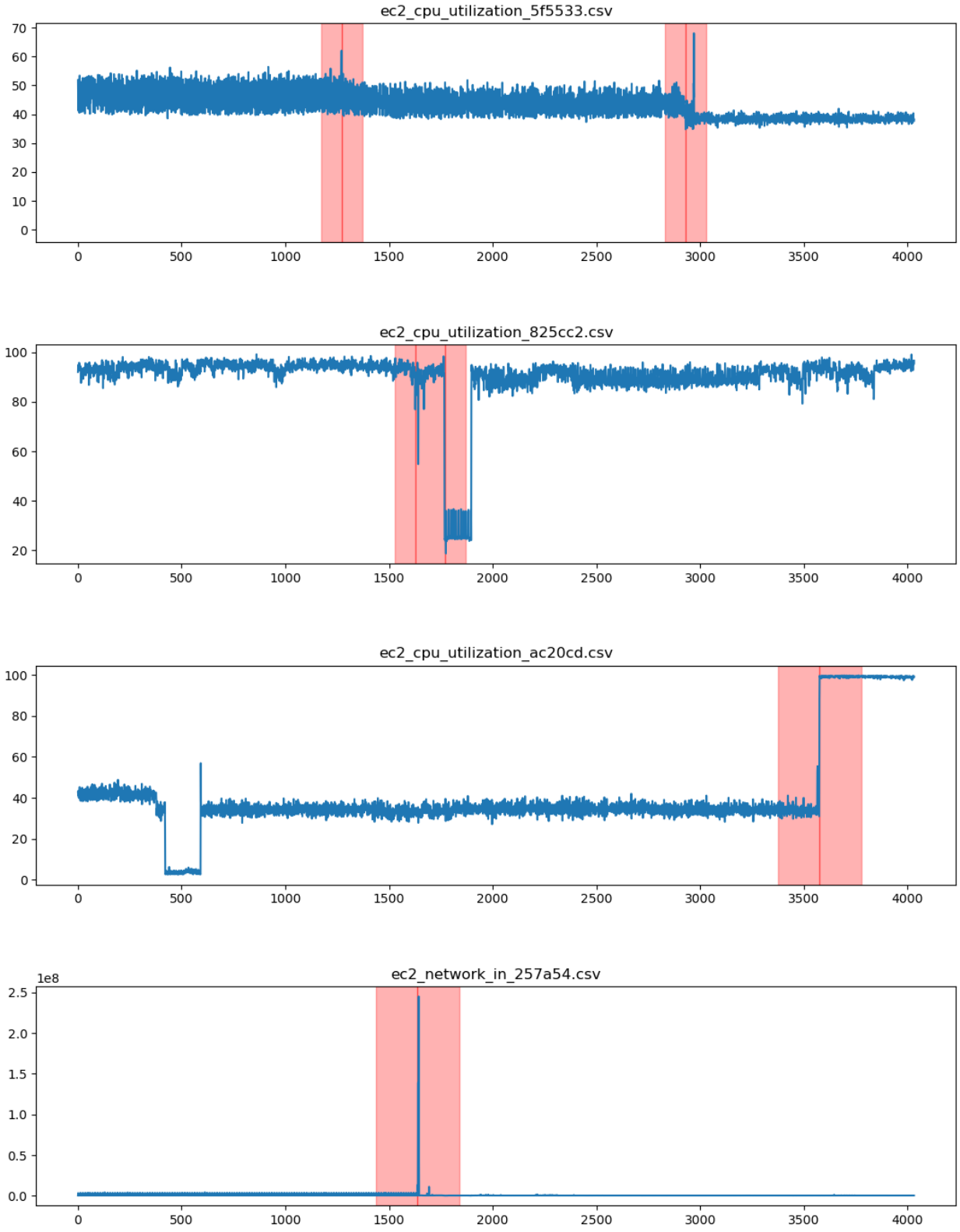


FIGURE A.1 Séries temporelles utilisées partie 1

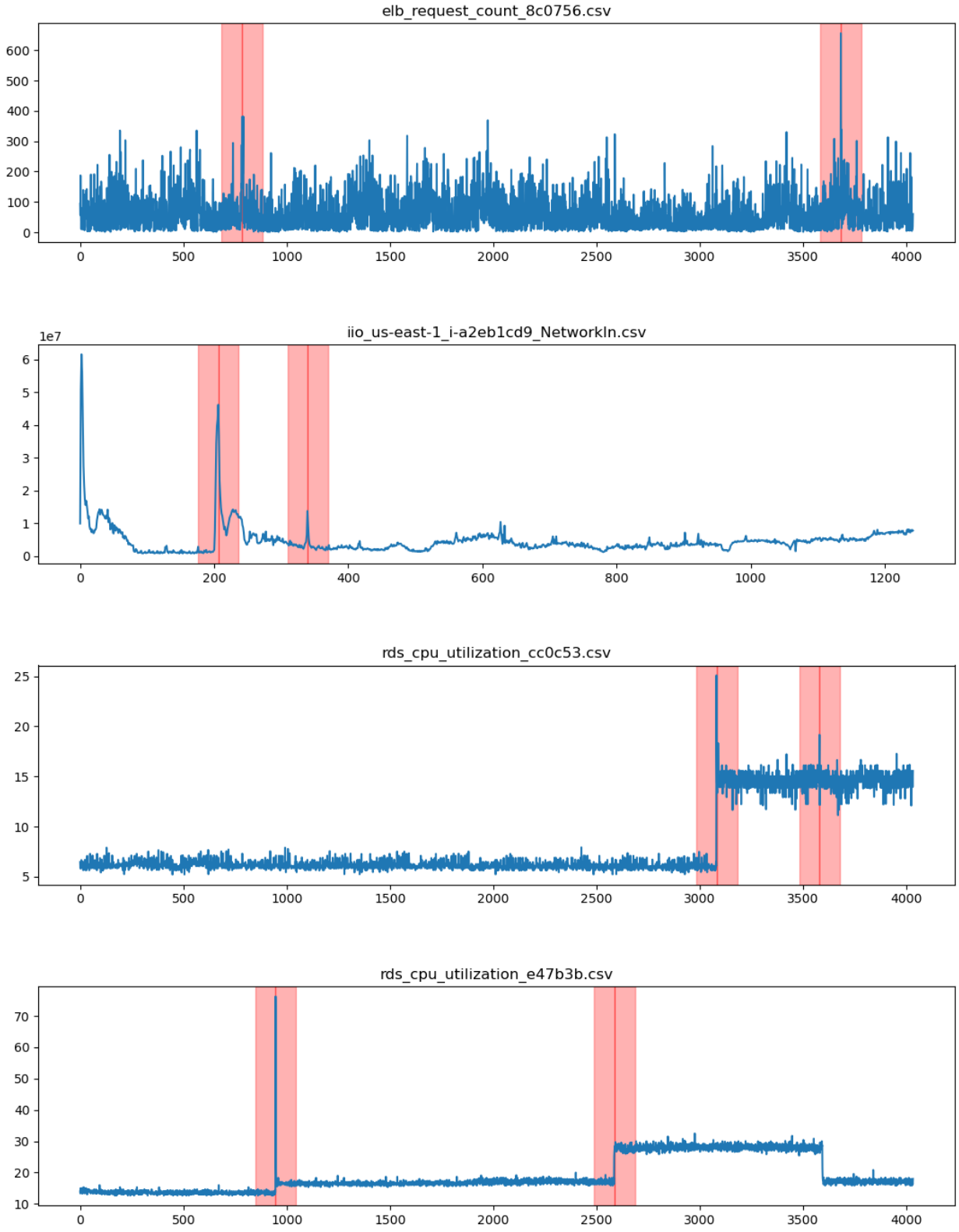


FIGURE A.2 Séries temporelles utilisées partie 2