Contents

I. INTRODUCTION

Today, classical machine learning is affecting the understanding and better regulation of the classical networking infrastructure underpinning the modern internet. This includes pattern recognition for individual computing devices, security and fault management, routing and traffic management, resource management, and distributed computation. In these applications, it is not only the speed of processing that is important, but reliability and security can also be paramount.

The large intersection between machine learning and network systems (the internet included) is perhaps not surprising. Firstly, machine learning relies on access to data and in many real-life applications, data naturally arises from distributed sources. Secondly, especially for complex systems like large networks, the information to process is complex, containing many uncertainties, and subject to errors. Exactly solvable models in these regimes are few and far between, and machine learning may be helpful in making predictions in these complex, messy environments.

In the coming quantum era we can envision three distinct ways where quantum resources might be introduced: quantum communication; quantum processing at individual nodes; and, data that is inherently quantum in nature. To begin, we first make a classification of the four different foreseeable types of networks, see Tab. I. These can be classified as:

- CC: classical data and processing over a classical network (e.g the present-day internet).
- CQ: classical data and processing over a quantum network.
- QC: quantum data and processing over a classical network.
- QQ: quantum data and processing over a quantum network – a fully-quantum internet.

Now we can pose the inevitable question: how do these different quantum resources in a network relate to machine learning?

This is not yet an active research area in its own right. However, there are some preliminary toolkits we can consider that are starting to be developed in the new field of quantum machine learning. By first summarising the intersection between the classical internet and classical machine learning, we gain insight into the kinds of tools required to start examining their quantum counterparts. For instance, we will see how aspects of quantum processing of quantum data might be aided by machine learning, how machine learning may be enhanced by quantum resources, and also in turn how machine learning may be implemented in these distributed quantum settings. See Tab. II for a brief overview.

II. CLASSICAL MACHINE LEARNING IN CLASSICAL NETWORKS

To create and maintain efficient classical networks, one requires efficient and reliable processing at individual nodes, secure processing, efficient routing and data transmission, efficient uses of resources, and a means for distributed information processing. For an overview of methods see for example (Boutaba *et al.*, 2018; Wang *et al.*, 2018). We present a brief overview on how machine learning can be used in each of these areas.

A. Machine learning basics for individual processors

Machine learning algorithms allow us to make predictions about a current or future dataset without requiring explicit instructions. Since its aims are directed more towards *prediction* rather than purely *estimation*, it differs from the field of statistical estimation, although it shares many techniques in common.

There are three mains paradigms for machine learning:

- Supervised learning: relies upon training data from which inferences and predictions about new test data can be extracted.
- Unsupervised learning: makes inferences from the data at hand without access to a training stage, much like a student learning without an instructor.
- Reinforcement learning: operates using a different framework and aims to find the best action steps in a particular environment to maximise a given reward.

Machine learning is used regularly for data collection, feature engineering, and model learning. There are many excellent introductory texts on this topic (Bishop, 2006; Flach, 2012; Marsland, 2011; Shalev-Shwartz and Ben-David, 2014; Trevor *et al.*, 2009).

B. Machine learning for security and fault management

There are two main ways in which machine learning enters into managing security and faults in networks. The first is *using* machine learning techniques to predict and detect security breeches and faults in a network. These include machine learning algorithms for anomaly detection. The second is studying the security vulnerabilities of machine learning algorithms themselves, as the presence of adversaries becomes natural in a network setting, where real-life machine learning algorithms will be deployed. The latter is known as adversarial machine learning.

1. Anomaly detection and fault management

When there are security breeches in a classical network, one desires the ability to predict and detect these disturbances, as well as a method for making one's protocols more robust against adversaries. Machine learning is often used in anomaly detection and intrusion detection. These algorithms look for unusual data or changes in the data. Broadly, there are three classes of anomalies: point, contextual, and collective, which refer respectively to single datum anomalies, unusual data with respect to a specified context, and clusters of data which together point to an unusual pattern. Both supervised and unsupervised algorithms are used in these settings (Ahmed *et al.*, 2007; Thottan and Ji, 2003). One of the prime challenges here is determining the presence of an anomaly when little data is available, and determining the relevant rate of false positives and negatives for a particular application.

Fault management in a network is also extremely relevant, especially for complex networks where there is more room for errors. One requires the prediction, detection and localisation of the fault, and most relevant machine learning methods use supervised algorithms. However, the paucity of real training data (as opposed to synthetic data generated from simulations) means that the algorithms might be poorly trained, especially in newly established networks (Hood and Ji, 1997; Kogeda and Agbinya, 2006; Snow *et al.*, 2005). This is to be reasonably anticipated with the deployment of the future quantum internet. To accommodate for this, new methods have arisen where unsupervised machine learning techniques are used instead to detect changes in the network rather than relying on labelled fault data (Hajji, 2005).

In particular, to identify and localise unusual behaviour in a network, which can be due to natural faults or an adversarial party, network anomaly detection methods can be employed (Ahmed *et al.*, 2007; Fraley and Cannady, 2017; Joseph *et al.*, 2013). Since the results can be sensitive to the training datasets used, it is important to examine which datasets are most appropriate for one's particular applications [see (Yavanoglu and Aydos, 2017) for a review]. In particular, there have been many proposals for utilising anomaly detection in network intrusion. However, this approach has been criticised for its use in real-world applications, where it's often difficult to distinguish anomalies related to intrusions from those attributed to other factors, and the complexity of real-world networks may make it too difficult to define what a normal signal is (Sommer and Paxson, 2010).

2. Adversarial machine learning

Machine learning algorithms themselves are vulnerable to security attacks. This comes under the field of adversarial machine learning (Huang *et al.*, 2011). There are two main types of attacks:

- Evasion: directed at the test data.
- Poisoning: directed at the training data and machine learning models.

In real-life scenarios, data often originates from different sources, making adversarial attacks more likely. It has been discovered that many machine learning algorithms are in fact vulnerable to adversarial attacks, the first discovered in (Szegedy *et al.*, 2013). A large proportion of the literature focuses on the details of specific algorithms: the detection of adversaries; their different methods of attack; and, the particular defences against them (Kurakin *et al.*, 2018). However, recently, more foundational work has emerged, explaining the origins of this vulnerability as arising from the high dimensionality of the underlying data (Gilmer *et al.*, 2018; Goodfellow *et al.*, 2014; Mahloujifar *et al.*, 2018).

C. Machine learning for traffic and routing management

The effective operation of a large-scale network also requires automated management. This includes efficient

means for traffic prediction, traffic classification, routing, and congestion control. Machine learning algorithms have been developed for all of these.

1. Traffic

Predicting network traffic is becoming increasingly important, especially in diverse and complex networks. This is commonly addressed using time-series forecasting (TSF) methods. This can make use of either statistical analysis models or supervised machine learning methods (Bermolen and Rossi, 2009; Chabaa *et al.*, 2010; Cortez *et al.*, 2006). Non-TSF methods also exist (Chen *et al.*, 2016; Li *et al.*, 2016).

The most commonly-used technique for traffic classification is the so-called flow feature-based technique. This takes into account information on unidirectional packets sent across the network. For this, supervised machine learning has been found to be accurate for traffic classification. However, unsupervised techniques have been found to be more robust. Their joint application is a very powerful technique (Erman *et al.*, 2007; Zhang *et al.*, 2015).

2. Routing

Machine learning is most applicable to dynamic routing problems, requiring rapid updating of optimal routes. Otherwise, existing routing tables can be sufficient. Since these settings depend on frequent recalculation, reinforcement learning algorithms are the most appropriate. In particular, Q-learning has performed well in various networks (Forster and Murphy, 2007; Wang and Wang, 2006; **?**).

3. Congestion

Congestion control in a network is important to ensure stability and the minimisation of packet loss. Well-known congestion control methods like queue management already exist. However, machine learning can be used to enhance the effectiveness of congestion control in various scenarios, especially for TCP/IP networks (Barman and Matta, 2004; El Khayat *et al.*, 2005; Liu *et al.*, 2002).

D. Distributed machine learning

Distributed machine learning is simply the fusion of distributed computation and machine learning, where the learning algorithm is implemented across a network in a distributed fashion. They become highly relevant in the following scenarios:

- Training and/or testing data originates from different sources. This is the naturally-distributed setting.
- There is too much data to store locally on a single device.
- When fault tolerance becomes important. For instance, for high-value data, decentralised storage provides enhanced integrity.

The toolbox and infrastructure for distributed machine learning is rapidly developing, and there are many known algorithms (Florina Balcan and Liang, 2013; Peteiro-Barral and Guijarro-Berdiñas, 2013). Existing platforms catering for distributed machine learning include MLbase (Kraska *et al.*, 2013), Hadoop (White, 2012), and Spark (Shanahan and Dai, 2015).

Caution is required, however, as there are cases when one *shouldn't* employ distributed machine learning, such as when:

- Communication and synchronisation between the distributed parties provides a bottleneck for the computation.
- Writing and running a distributed program is too complicated.
- One can run the same algorithm on a multi-core machine. This is possible with smart data sampling, offline schemes, and efficient parallel codes.

III. CLASSICAL DATA AND MACHINE LEARNING WITH QUANTUM RESOURCES

There are at least three broad ways in which we can employ quantum resources for classical data over a network:

- Quantum resources enhance data-processing at individual nodes.
- Quantum resources improve security.
- Quantum resources enhance communication.

In a classical network, the first question is whether or not quantum resources can assist in any of the relevant machine learning algorithms, including improving security. These belong to the class of quantum-enhanced machine learning algorithms.

In a quantum network with only classical data, communication complexity improvements are possible (Brassard, 2003). It is unclear whether machine learning has utility in this setting, although there are some promising clues (Balcan *et al.*, 2012; Conitzer and Sandholm, 2004; Kane *et al.*, 2017) on the connection between communication complexity and machine learning.

A. Quantum-enhanced machine learning overview

Quantum-enhanced machine learning algorithms are quantum algorithms performing machine learning tasks. They have so far mostly concentrated on quantum speed-ups with respect to the dimensionality of the underlying data.

1. Fully-quantum algorithms

The first of these algorithms chiefly relied on assuming completely quantum devices, maintaining coherence throughout the computation, and could require full fault-tolerance. For algorithms claiming exponential quantum ehnhancement for supervised learning (Biamonte *et al.*, 2017; Ciliberto *et al.*, 2018), the HHL algorithm (Harrow *et al.*, 2009) for matrix inversion is often employed. However, HHL has a number of drawbacks, making it impractical for near-term quantum devices:

- The ability to efficiently encode classical data into quantum states and quantum memory (Aaronson, 2015).
- Effective quantum state read-out (Aaronson, 2015).
- They generally require high circuit-depth.
- There are restrictions on the sparsity and conditioning of matrices.

Although later developments have tried to circumvent restrictions on sparsity and rather focus on low-rank matrices (e.g quantum principal component analysis for low-rank matrices (Lloyd *et al.*, 2014)), recent work on quantum-inspired classical algorithms has demonstrated efficient classical algorithms can exist in these cases (Chia *et al.*, 2018; Gilyén *et al.*, 2018; Tang, 2018). In fact, classical sampling methods developed in (Tang, 2018) for quantum-inspired machine learning algorithms suggest that classical methods for linear algebra problems in low-dimensions (used in machine learning for instance) are likely to find efficient classical algorithms. Although these classical sampling methods are not yet more practical than existing classical sampling methods, they are still more realistic than their quantum counterparts.

Another set of approaches, relying on amplitude amplification and Grover's search algorithm, can provide up to quadratic runtime enhancement. These include quantum algorithms for reinforcement learning (Dunjko *et al.*, 2016) and training of quantum perceptrons (Kapoor *et al.*, 2016). While theoretically very interesting as long-term goals, near-term proposals are missing.

2. Hybrid algorithms

To find algorithms that may be realised in the the near-term, quantum machine learning algorithms are now dedicating more attention to hybrid classical-quantum algorithms. These algorithms, which include variational methods for optimisation (Moll *et al.*, 2018), exhibit low circuit-depth, where the optimisation process is performed iteratively and classically. They are of roughly two types: one that attempts to enhance classical algorithms with classical input data, and another where the quantum advantage lies in efficient quantum state preparation, thus using quantum input data. Prominent examples of the former include quantum approximate optimisation algorithms (QAOA) (Farhi *et al.*, 2014; Farhi and Harrow, 2016) and the latter includes variational quantum eigensolvers (VQE) (Kandala *et al.*, 2017; Peruzzo, 2014). We return to VQE in the following section as these solve problems for quantum data.

Both QAOA and VQE can be considered as belonging to the same framework, and their optimisation component (which can be considered only as a component and not the entirety of machine learning) is performed classically. One begins with an ansatz quantum state. A unitary with classically-tuneable parameters is then applied to this state and an observable whose expectation value representing the cost function for the problem is subsequently measured. The classical parameters of the unitary are then iteratively tuned until a minimum in the cost function is reached (i.e a Hamiltonian ground state), for instance using the classical gradient-descent algorithm.

In QAOA, the ground state reached then encodes the classical solution to a classical optimisation problem, like MaxCut, with polynomial runtime. Thus, it is not a quantum-enhanced algorithm for a classical machine learning problem, but rather exploits a classical machine learning algorithm. It remains to be seen if optimisation problems more directly relevant to networking applications can be solved in this way.

Alternate frameworks have been developed to find quantum-enhanced algorithms that not only take advantage of classical optimisation algorithms, but also enhance classical machine learning algorithms. These new proposals include quantum circuit learning (Mitarai *et al.*, 2018), quantum generalisation of neural networks (**?**), and Born machines (Benedetti *et al.*, 2018; Cheng *et al.*, 2018). Theoretical demonstration of quantum enhancement in these settings remains an open problem.

B. Quantum-enhanced machine learning for security and other applications

1. Anomaly detection

The chief machine learning method for detecting and averting faults and security breaches in classical networks is in anomaly detection. However, for anomalies in classical data, it appears unlikely that currently available quantum machine learning algorithms can enhance detection speed and reliability. One of the primary reasons

is the necessity for encoding classical data into quantum states, which can be very costly (Aaronson, 2015). Thus, even if there are quantum-enhanced supervised and unsupervised machine algorithms for anomaly detection in the computational component, the state preparation and readout demands may be excessive. However, this is no longer the case if we begin with quantum data instead, to be discussed in Sec. IV.

2. Adversarial quantum machine learning

Just as machine learning algorithms are vulnerable to attacks, so are quantum-enhanced machine learning ones. This is a very new area, known as *adversarial quantum machine learning*. As with adversarial machine learning, the aim is to find more robust quantum machine learning algorithms, and some robust algorithms have indeed been proposed (Wiebe and Kumar, 2018). In addition to finding more robust algorithms, it is also important to understand what the limitations on robustness of quantum machine learning algorithms are, currently an open problem. A recent result suggests that the same quantum resource requirements may be necessary for detecting adversaries in higher dimensions as compared to quantum tomography (**?**). Thus, it remains unclear the total resource cost of quantum-enhanced machine learning in the presence of adversaries. However, there is a tantalising yet unexplored possibility that quantum resources may enhance the security of machine learning algorithms, in a similar way that information-theoretic security is afforded by quantum cryptographic protocols.

3. Other

Whether or not there are helpful applications of quantum-enhanced machine learning algorithms for traffic and routing management is currently very unclear, and may even appear unlikely. There may be some quantum-enhancements to supervised and unsupervised machine algorithms that could be used in traffic and routing management. However, a key issue still remains in how classical data can be embedded into the relevant quantum states and then read out, in a way that is easier in the quantum setting. The no-cloning theorem forbids reproducing the state, and in general overheads in embedding classical information into quantum states are very high (Giovannetti *et al.*, 2008a,b). Additionally, given the dynamic nature of networks, where machine learning methods appear to be most helpful, the speed of embedding classical data into quantum states must be similarly high. Thus, the necessary quantum resources only to convert classical data to quantum states may overwhelm computational advantage.

C. Distributed quantum machine learning

The motivations for considering distributed quantum machine learning are similar to those for distributed classical machine learning. Suppose one wishes to perform distributed machine learning, either because the given data is naturally distributed or there is limited processing power on any given device. For this purpose, there are existing protocols for implementing general distributed quantum algorithms that could also be helpful in delegating quantum machine learning algorithms (Beals *et al.*, 2013).

Secure delegated quantum computational protocols (Joseph F. Fitzsimons, 2017) can also be modified and applied to quantum machine learning (Bang *et al.*, 2015; Sheng and Zhou, 2017). However, the same problem with state preparation could persist, for the server rather than the client. Alternatively, hybrid-classical quantum algorithms for distributed quantum machine learning have been developed in (Yoo *et al.*, 2014). Here, the quantum state preparation assumptions can be obviated by using a hybrid gate that takes in classical input data and implements classically-controlled unitary evolution.

IV. QUANTUM DATA AND MACHINE LEARNING

Suppose our data is inherently quantum in the form of quantum states or channels – *quantum data*. We may also face restrictions in the number of copies we have access to, imposed by the no-cloning theorem.

In these cases, it is found that classical machine learning methods may be helpful over traditional methods in dealing with quantum data. Another approach is to use quantum protocols to directly process quantum data. Learning protocols in the latter case belong to the field of quantum learning.

It is possible to process quantum data over both classical and quantum networks. Techniques from classical machine learning for quantum data may assist in the communication of quantum data over a classical network, while quantum learning protocols may be more appropriate over a quantum network. This is a new research direction, and it is presently unclear exactly if and how these methods may be applicable, tantalising inspiration for future study.

A. Classical machine learning for quantum data

1. Tomography

Suppose data is naturally in the form of quantum states or channels. Then for classical processing over a classical network, the first task is to find its classical description. The canonical methods for this are quantum state tomography and quantum process tomography. However, tomography is in general very resource

intensive. Recent work has presented methods for efficient state tomography using classical machine learning techniques (Torlai *et al.*, 2018; Wang *et al.*, 2017) over a larger range of states than previously studied.

2. Separability

While tomography provides complete classical descriptions of quantum data, sometimes it may be sufficient to first classify the data in terms of quantum properties. For instance, methods for classifying quantum states directly in terms of separability have been devised using classical machine learning (Gao *et al.*, 2018; Lu *et al.*, 2018; Ma and Yung, 2017). Here there are empirical demonstrations of some advantages compared to the CHSH inequality. However, accumulating sufficient training data may still remain a problem for high-dimension states.

3. Automated experiment design

In a future quantum internet with entangled quantum networks, it is desirable to find optimal methods for generating inter-node entanglement. It is also desirable for this process to an automated one. Recently, such automated methods based on classical reinforcement learning (Alexey A Melnikov, 2018) have been proposed to experimentally create a variety of entangled states with greater efficiency. This provides an exciting starting point for automated design in future quantum internet protocols.

4. Variational quantum eigensolvers

We saw that variational quantum eigensolvers (VQE) rely on classical optimisation. When applied to quantum data, they have found success mostly in quantum chemistry (Moll *et al.*, 2018; Peruzzo, 2014). In the context of quantum networks, the most promising developments are perhaps in its applicability to quantum data compression (**?**), which may assist in the quantum date communication.

B. Quantum learning protocols

1. Template matching

The first quantum algorithms for quantum data most relevant for machine learning are quantum template matching algorithms (Masahide Sasaki, 2002; **?**). These are classification algorithms, where each class is represented by a quantum state: a 'template'. The task is to find the class to which a given test quantum state belongs, where this state is not identical to any of the template states. It is not clear on its own whether quantum template matching is directly useful in a quantum network setting. However, the ideas introduced here provide the key foundation for supervised learning of quantum data, which can be used in the quantum counterparts to supervised algorithms in traffic prediction, classification and anomaly detection.

2. Learning quantum processes

Suppose one wants to send just enough information about a quantum process over a quantum network in order for the other parties to replicate the use of this process onto any desired quantum state. In the quantum data scenario, one is not a priori given the classical description of this quantum process. Instead, one is provided only a finite number of queries of this quantum process. For a unitary operation, this problem is tackled in (**?**), in a problem called the quantum learning of unitary operations. A very interesting observation made here is that the optimal strategy is semi-classical instead of fully-quantum: meaning that it is sufficient for the classical data encoding the estimation of the unknown unitary to be stored. It remains an open question on how these results may change if one extends to more general quantum processes.

3. Quantum learning and security

In a future quantum network with quantum data being exchanged between different parties, it becomes important to detect unusual behaviour in the incoming quantum data. These may be the first signs of a security breach or a fault in the network. For dynamical data in a time-series, for instance, change point detection addresses precisely this problem. This has been extended to the quantum domain (Gael Sentís, 2016; Shang Yu, 2018) where the optimal methods for detecting a change in quantum data are found using methods from state discrimination. For static data, anomaly detection methods based on machine learning become more appropriate as the definition of of unusual behaviour is based on priori training data. Classical anomaly detection algorithms have been applied to quantum data in (Satoshi Hara, 2014) for the purpose of error detection, in the case where the classical description for the quantum data is known. However, for cases where the classical description for the quantum data is unknown (as expected over a quantum internet), it is instead much more efficient to directly apply quantum algorithms onto the quantum data directly. Examples of this include several quantum algorithms for anomaly detection was proposed in (Liu and Rebentrost, 2018).

References

Aaronson, S., 2015, Nature Physics **11**, 291.

Data/Network	Classical	Quantum
Classical	Current internet (CC)	Classical data in quantum network (CQ)
Quantum	Quantum data in classical network (QC)	Fully quantum internet (QQ)

Table I Classical and quantum data in a network

Network concerns	Classical data	Quantum resources and classical data	Quantum data
Individual computing	Classical machine learning	Quantum-enhanced machine learning	Quantum learning
Security and faults	Machine learning for anomalies and faults: detection and prediction; Adversarial machine learning	Adversarial quantum machine learning	Anomaly detection and change point detection for quantum data
Routing and traffic	Machine learning for traffic prediction, classification, congestion control and routing	Open problems	Open problems
Distributed computing	Distributed machine learning	Distributed quantum machine learning	Distributed quantum learning
Communication	Data compression and machine learning	Open problems	Data compression and quantum machine learning

Table II Classical and quantum machine learning applications in classical and quantum networks. Almost all of the categories here are very new and open to exploration in the quantum domain.

Ahmed, T., B. Oreshkin, and M. Coates, 2007, in *Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques*, p. 1.

Alexey A Melnikov, M. K. V. D. M. T.-A. Z. H. J. B., Hendrik Poulsen Nautrup, 2018, Proceedings of the National Academy of Sciences **115**, 1221.

Balcan, M. F., A. Blum, S. Fine, and Y. Mansour, 2012, in *Conference on Learning Theory*, p. 26, eprint arXiv:1204.3514v3.

Bang, J., S.-W. Lee, and H. Jeong, 2015, Quantum Information Processing **14**, 3933.

Barman, D., and I. Matta, 2004, in *Proceedings of WiOpt*, volume 4.

Beals, R., S. Brierley, O. Gray, A. W. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather, 2013, Proceedings of the Royal Society A **469**, 20120686.

Benedetti, M., D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, 2018, eprint arXiv:1801.07686.

Bermolen, P., and D. Rossi, 2009, Computer Networks **53**(2), 191.

Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, 2017, Nature **549**, 195.

Bishop, C. M., 2006, *Pattern recognition and machine learning* (Springer).

Boutaba, R., M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, 2018, Journal of Internet Services and Applications **9**, 16.

Brassard, G., 2003, Foundations of Physics **33**, 1593.

Chabaa, S., A. Zeroual, and J. Antari, 2010, Journal of Intelligent Learning Systems and Applications **2**, 147.

Chen, Z., J. Wen, Y. Geng, *et al.*, 2016, in *IEEE 24th International Conference on Network Protocols (ICNP)*, p. 1.

Cheng, S., J. Chen, and L. Wang, 2018, Entropy **20**, 583.

Chia, N.-H., H.-H. Lin, and C. Wang, 2018, eprint arXiv:1811.04852.

Ciliberto, C., M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, 2018, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **474**, 20170551.

Conitzer, V., and T. Sandholm, 2004, in *ACM Proceedings of the twenty-first international conference on Machine learning*, p. 24.

Cortez, P., M. Rio, M. Rocha, and P. Sousa, 2006, in *IEEE International Joint Conference on Neural Networks (IJCNN'06)*, p. 2635.

Dunjko, V., J. M. Taylor, and H. J. Briegel, 2016, Physical Review Letters **117**, 130501.

El Khayat, I., P. Geurts, and G. Leduc, 2005, in *International Conference on Research in Networking*, p. 549.

Erman, J., A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, 2007, Performance Evaluation **64**, 1194.

Farhi, E., J. Goldstone, and S. Gutmann, 2014, eprint arXiv:1411.4028.

Farhi, E., and A. W. Harrow, 2016, eprint arXiv:1602.07674.

Flach, P., 2012, *Machine learning: the art and science of algorithms that make sense of data* (Cambridge University Press).

Florina Balcan, S. E., Maria, and Y. Liang, 2013, Advances in Neural Information Processing Systems , 1995eprint arXiv:1306.0604v3.

Forster, A., and A. L. Murphy, 2007, in *IEEE 3rd International Conference on Intelligent Sensors, Sensor Networks and Information (ISSNIP)*, p. 371.

Fraley, J. B., and J. Cannady, 2017, in *IEEE SoutheastCon*, p. 1.

Gael Sentís, J. C. G. C. R. M.-T., Emilio Bagan, 2016, Physical Review Letters **117**, 150502.

Gao, J., L.-F. Qiao, Z.-Q. Jiao, Y.-C. Ma, C.-Q. Hu, R.-J. Ren, A.-L. Yang, H. Tang, M.-H. Yung, and X.-M. Jin,

2018, Physical Review Letters **120**.
Gilmer, J., L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow, 2018, eprint arXiv:1801.02774.
Gilyén, A., S. Lloyd, and E. Tang, 2018, eprint arXiv:1811.04909.
Giovannetti, V., S. Lloyd, and L. Maccone, 2008a, Physical Review A **78**, 052310.
Giovannetti, V., S. Lloyd, and L. Maccone, 2008b, Physical Review Letters **100**, 160501.
Goodfellow, I. J., J. Shlens, and C. Szegedy, 2014, eprint arXiv:1412.6572.
Hajji, H., 2005, IEEE Transactions on Neural Networks **16**, 1053.
Harrow, A. W., A. Hassidim, and S. Lloyd, 2009, Physical Review Letters **103**, 150502.
Hood, C. S., and C. Ji, 1997, IEEE Transactions on reliability **46**, 333.
Huang, L., A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, 2011, in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, p. 43.
Joseph, A. D., P. Laskov, F. Roli, J. D. Tygar, and B. Nelson, 2013, in *Dagstuhl Manifestos*, volume 3.
Joseph F. Fitzsimons, E. K., 2017, Physical Review A **96**, 012303.
Kandala, A., A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, 2017, Nature **549**, 242.
Kane, D. M., R. Livni, S. Moran, and A. Yehudayoff, 2017, eprint arXiv:1711.05893.
Kapoor, A., N. Wiebe, and K. Svore, 2016, in *Advances in Neural Information Processing Systems*, p. 3999, eprint arXiv:1602.04799v1.
Kogeda, P., and J. I. Agbinya, 2006, in *International conference on Wireless Broadband and Ultra Wideband Communication* (UTS ePress).
Kraska, T., A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan, 2013, Cidr **1**, 2.
Kurakin, A., I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, *et al.*, 2018, in *The NIPS'17 Competition: Building Intelligent Systems* (Springer), p. 195, eprint arXiv:1804.00097v1.
Li, Y., H. Liu, W. Yang, D. Hu, and W. Xu, 2016, in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, p. 206.
Liu, J., I. Matta, and M. Crovella, 2002, *End-to-end inference of loss nature in a hybrid wired/wireless environment*, Technical Report, Boston University Computer Science Department.
Liu, N., and P. Rebentrost, 2018, Physical Review A **97**, 042315.
Lloyd, S., M. Mohseni, and P. Rebentrost, 2014, Nature Physics **10**, 631.
Lu, S., S. Huang, K. Li, J. Li, J. Chen, D. Lu, Z. Ji, Y. Shen, D. Zhou, and B. Zeng, 2018, Physical Review A **98**, 012315.
Ma, Y.-C., and M.-H. Yung, 2017, eprint arXiv:1705.00813.
Mahloujifar, S., D. I. Diochnos, and M. Mahmoody, 2018, eprint arXiv:1809.03063.
Marsland, S., 2011, *Machine learning: an algorithmic perspective* (Chapman and Hall/CRC).
Masahide Sasaki, A. C., 2002, Physical Review A **66**, 022303.
Mitarai, K., M. Negoro, M. Kitagawa, and K. Fujii, 2018, Physical Review A **98**, 032309.
Moll, N., P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, *et al.*, 2018, Quantum Science and Technology **3**, 030503.
Peruzzo, A., 2014, Nature Communications **5**, 4213.
Peteiro-Barral, D., and B. Guijarro-Berdiñas, 2013, Progress in Artificial Intelligence **2**, 1.
Satoshi Hara, R. O. T. W. S. T., Takafumi Ono, 2014, Physical Review A **89**, 022104.
Shalev-Shwartz, S., and S. Ben-David, 2014, *Understanding machine learning: From theory to algorithms* (Cambridge University Press).
Shanahan, J. G., and L. Dai, 2015, in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, p. 2323.
Shang Yu, J.-S. T. Z.-A. J. Y.-T. W. Z.-J. K. W. L. X. L. Z.-Q. Z. Z.-D. C. J.-S. X. Y.-C. W. Y.-Y. Z. G.-Y. X. C.-F. L. G.-C. G. G. S. R. M.-T., Chang-Jiang Huang, 2018, Physical Review A **98**(4), 040301.
Sheng, Y.-B., and L. Zhou, 2017, Science Bulletin **62**, 1025.
Snow, A., P. Rastogi, and G. Weckman, 2005, in *IEEE Military Communications Conference (MILCOM)*, p. 2809.
Sommer, R., and V. Paxson, 2010, in *IEEE Symposium on Security and Privacy (SP)*, p. 305.
Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, 2013, eprint arXiv:1312.6199.
Tang, E., 2018, eprint arXiv:1811.00414.
Thottan, M., and C. Ji, 2003, IEEE Transactions on signal processing **51**, 2191.
Torlai, G., G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, 2018, Nature Physics **14**, 447.
Trevor, H., T. Robert, and F. JH, 2009, The elements of statistical learning: data mining, inference, and prediction.
Wang, J., Z.-Y. Han, S.-B. Wang, Z. Li, L.-Z. Mu, H. Fan, and L. Wang, 2017, eprint arXiv:1712.03213.
Wang, M., Y. Cui, X. Wang, S. Xiao, and J. Jiang, 2018, IEEE Network **32**, 92.
Wang, P., and T. Wang, 2006, in *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, p. 219.
White, T., 2012, *Hadoop: The definitive guide* ("O'Reilly Media, Inc.").
Wiebe, N., and R. S. S. Kumar, 2018, New Journal of Physics eprint arXiv:1711.06652v1.
Yavanoglu, O., and M. Aydos, 2017, in *IEEE International Conference on Big Data*, p. 2186.
Yoo, S., J. Bang, C. Lee, and J. Lee, 2014, New Journal of Physics **16**, 103014.
Zhang, J., X. Chen, Y. Xiang, W. Zhou, and J. Wu, 2015, IEEE/ACM Transactions on Networking (TON) **23**, 1257.