POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Evaluation of quantum machine learning algorithms for cybersecurity

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Tommaso Fioravanti

Advisor: Prof. Stefano Zanero

Co-advisors: Armando Bellante Michele Carminati Alessandro Luongo

Academic year: 2021-2022

1. Introduction

Quantum machine learning is a novel discipline that combines concepts from quantum computing and machine learning, providing speed-ups and improved performances over classical solutions. Although both quantum machine learning and, more generally, quantum computing are still studied a lot on a theoretical level, they are very promising in many fields. Indeed, it is clear that quantum technology will have a big impact on computer security tasks by having already greatly affected cryptography. As a matter of fact, nowadays, cybersecurity experts search for *post-quantum* cryptography algorithms which are algorithms resistant also to quantum attacks [4, 10]. It is important that cryptographers start now to build new quantum-proof algorithms before the advent of the quantum computers era. This is because it has been estimated that migrating to post-quantum cryptography will take about fifteen years which is practically the estimated time for the advent of large-scale quantum computers [11]. It is important not to be caught unprepared by the arrival of quantum computers since a quantum-enabled attacker can break a cryptosystem whose security relies on the difficulty to find solutions to hard mathematical problems with a classical computer. For instance, *Shor*'s algorithm can break RSA, ECC, and DH algorithms as well as *Grover*'s algorithm jeopardizes the security of hash algorithms, such as SHA, and symmetric key algorithms such as AES.
So, it is evident that quantum computing changes the dynamics of cryptography. However, cryptography is not the only field in cybersecurity. Indeed, there are many other cybersecurity tasks such as intrusion detection or malware detection. In these tasks, machine learning is widely used both to perform attacks and to automate defense systems. That said, and given that quantum machine learning is increasingly popular, we try to figure out if and how quantum machine learning can affect cybersecurity. In this work, we begin to understand if it is reasonable to think about possible applications of quantum machine learning in future cybersecurity.

1.1. Main contributions

- We develop an open-source framework to simplify the process of simulation of quantum algorithms as we were using a fault-tolerant quantum computer.
- We find some interesting insights of tomography routine which we exploit in its application in quantum machine learning algorithms that we use to solve intrusion detection problems.

- We extend a PCA-based anomaly detection algorithm, improving its performances over the CICIDS dataset. We report an F1-score improvement of $\approx 15\%$ and an accuracy improvement of $\approx 10\%$.
- We report a study of the error of quantum machine learning algorithms to see how it affects the performances of quantum intrusion detection models.
- We report a critical analysis of the running times of quantum machine learning models compared to the classical ones.

2. Approach

We apply quantum machine algorithms to solve network intrusion detection problems as we were using a fault-tolerant quantum computer. We consider quantum machine learning algorithms as randomized approximation algorithms. Indeed, they are formalized in theorems and proofs where the running time, the probability of failure γ, and the approximation error ϵ are specified. In these algorithms, we might not know exactly the real solution s of our problem but we can approximate it with an absolute error such that $\|s - \overline{s}\|_2 \leq \epsilon$ with a probability of $1 - \gamma$. The approximation error is expected by the quantum computational paradigm and it is not due to technological limits since we assume fault-tolerant quantum computers. The probability of failure γ and the approximation error ϵ appear as running time parameters. In this work, when we talk about simulating quantum algorithms, we do not intend to execute quantum circuits but rather simulate the approximation error. We insert ϵ error in the correct steps, as described in the proofs of the corresponding algorithms, to obtain estimates ϵ-close to the exact solution, as guaranteed by the theoretical error bound. In this way, we see how the error, or more precisely the approximated solution of quantum machine learning algorithms used to solve intrusion detection problems, affects the accuracy in detecting intrusion with respect to the exact solution of classical machine learning algorithms. The core part of the work is based on finding a trade-off between approximation error and running time. We find cases of errors that we tolerate to match classical performances, we fix the corresponding parameters, and we evaluate the running time as the number of samples and features change. We see how many samples and features we need to observe an advantage in using quantum instead of classical machine learning algorithms, trying to understand in which cybersecurity tasks we deal with data of that size. To simplify the process of simulating quantum machine learning algorithms and studying their running time, we develop an open-source framework.

3. Simulation of the framework

We simulate phase estimation, consistent phase estimation, amplitude estimation, and state tomography before applying them in quantum machine learning algorithms to solve anomaly detection problems. We also make tests to verify their correctness, discovering interesting insights into tomography.

3.1. Vector state tomography test

Quantum pure state tomography is the process by which we can retrieve classical information about a quantum state through measurements. Given a quantum state $|\boldsymbol{x}\rangle$ for a unit vector $\boldsymbol{x} \in \mathbb{R}^d$, the tomography algorithm outputs an estimate $\overline{\boldsymbol{x}}$ such that $\|\boldsymbol{x} - \overline{\boldsymbol{x}}\|_2 \leq \delta$ by performing $N = \frac{36d \log d}{\delta^2}$ measurements [8]. We doubted that N was a large bound and, even with a number of measurements lower than N, we could obtain an estimate $\overline{\boldsymbol{x}}$ with error δ. We confirm our doubt with the experiment reported in Figure 1. The plot shows how many measures (x-axis) are necessary to get a vector estimate with a specific error (y-axis) both following the theoretical bound (blue curve) and actually performing the tomography (orange curve). As we can see, the shape of the two curves is almost equal. However, there is a big difference in the magnitude of the values. Indeed, the plot tells us that to get an estimate with an error of 0.05, we would need $\approx 10^5$ measures instead of $\approx 10^8$ which corresponds to the theoretical bound. By performing that number of measurements, we see that the error that we obtain in the orange curve is almost zero, meaning that we obtain an estimate very close to the true vector. The blue curve shows that in theory, to reach such an estimate, we would need $\approx 10^{11}$ measures.

This insight into tomography allows us to simulate tomography with δ error (such as 0.05 in the experiment) and obtain an estimate with an

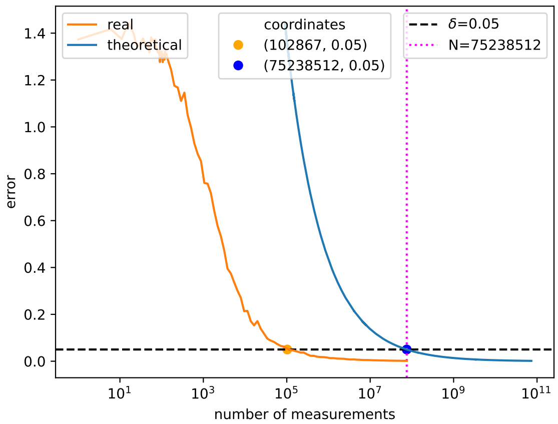

Figure 1: Comparison between theoretical and actual tomography. The dots indicate the number of measures for which we obtain estimates with error $\delta = 0.05$. The vertical dashed magenta line represents the number of measures given by $N = \frac{36d\log d}{0.05^2}$.

error much lower than δ (in this case almost zero as shown with the orange curve). We exploit this property in the simulation of tomography inside quantum machine learning algorithms. In this way, we can be less stringent on the error to be tolerated in the estimates by considering larger errors, thus impacting less on the running time of the tomography (since it scales as $\sim O\left(\frac{d}{\delta^2}\right)$) and still obtaining estimates closer to the real values with respect to the error δ.

4. Applications to cybersecurity

In this thesis, we simulate q-PCA and q-Means machine learning algorithms to solve network intrusion detection problems starting from already proposed classical anomaly detection models (we also extend them as described in Section 4.2). In this section, we report some of the most significant experiments by showing the comparison between classical and quantum in detecting intrusions.

Goals It is important to underline that our work is not meant to be a comparison between the intrusion detection models that we report and the state-of-the-art ones. Indeed, the goals of our experiments are basically two:

- Study how the error of quantum machine learning algorithms affects the accuracy of quantum models in detecting intrusions with respect to their classical counterparts.
- Compare running times of quantum and classical machine learning models to derive possible advantages in using one with respect to the other.

Datasets One of the main challenges in works concerning intrusion detection is the scarcity of real data to test models. We use three different publicly available datasets about network intrusion detection to fit and test our models: KDDCUP 99 [16], CICIDS 2017 [14], and DARKNET [6]. These datasets are composed of network packets labeled as "normal" or "attack". We do not distinguish between different types of attacks, but our goal is to build models that are able to differentiate normal from anomalous network traffic as accurately as possible.

4.1. Principal components classifier over KDDCUP

Model description The first model used as intrusion detection system is a PCA-based one and it is reported for the first time by Shyu et al. [15]. Given an input data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, the main concepts for this model are the two summations

$$T_1 = \sum_{i=1}^{k} \frac{y_i^2}{\lambda_i}, \quad T_2 = \sum_{i=d-r+1}^{d} \frac{y_i^2}{\lambda_i} \tag{1}$$

where k and r are the number of major (or principal) and minor components respectively and $y_i = \boldsymbol{v}_i^T\boldsymbol{z}$, with $\boldsymbol{z}$ vector of standardized observations and $\boldsymbol{v}_i$ i-th eigenvector corresponding to λ_i eigenvalue. We classify each observation $\boldsymbol{z}$ as an attack if

$$T_1 > c_1 \quad \textbf{or} \quad T_2 > c_2 \tag{2}$$

and normal otherwise. We define as c_1 and c_2 the oulier thresholds which are computed using T_1 and T_2 respectively and which are related to the false alarm rate α [15]. Basically, an increase of α corresponds to a decrease of the outlier threshold. For the quantum model, we use novel quantum machine learning algorithms to obtain classical estimates of $\boldsymbol{v}_i$ and λ_i such that $\|\boldsymbol{v}_i - \overline{\boldsymbol{v}}_i\|_2 \le \delta$ and $\|\lambda_i - \overline{\lambda}_i\|_2 \le 2\epsilon\sqrt{\lambda_i}$ in time $\widetilde{O}\left(\frac{\|\boldsymbol{X}\|\mu(\boldsymbol{X})kd}{\theta\sqrt{p}\epsilon\delta^2}\right)$ and $\widetilde{O}\left(\frac{\|\boldsymbol{X}\|\mu(\boldsymbol{X})k}{\theta\sqrt{p}\epsilon}\right)$ respectively

[2]. Once obtained these estimates, we are able to compute the two summations and perform the detection also for the quantum case.

Results We report the results of PCA70 and q-PCA70, which are models that retain 70% of the variance in the major components. In this experiment, we classify a sample as attack only if $\sum_{i=1}^{k} \frac{y_i^2}{\lambda_i} > c1$ and normal otherwise. We consider a training set of 5000 normal samples and a test set of 92,278 normal and 39,674 attack samples. In Table 1, we report the comparison between classical and quantum results. We report the results at the increase of the false alarm rate α. For the quantum experiment, we consider the following error parameters: $\epsilon_\theta = \epsilon = 1, \delta = 0.1$, and $\eta = 0.1$, where δ and ϵ are the errors in the estimate of singular vectors and singular values respectively, ϵ_θ and η are other two error parameters used in quantum routines necessary to perform q-PCA.

α	recall		precision	
	c	q	c	q
1%	0.9314	0.9284	0.9863	0.9868
2%	0.9319	0.9299	0.9818	0.9823
4%	0.9604	0.9575	0.9651	0.9657
6%	0.9851	0.9812	0.9420	0.9421
8%	0.9867	0.9836	0.9201	0.9207
10%	0.9944	0.9912	0.9005	0.9010

α	f1-score		accuracy	
	c	q	c	q
1%	0.9581	0.9567	0.9755	0.9747
2%	0.9562	0.9548	0.9743	0.9735
4%	0.9628	**0.9615**	0.9776	0.9769
6%	**0.9630**	0.9612	0.9772	0.9772
8%	0.9522	0.9511	0.9702	0.9696
10%	0.9451	0.9440	0.9653	0.9646

Table 1: Results comparison for classical (c) and quantum (q) principal components classifier with major components that retain 70% of the variance over KDDCUP.

As we can see, with the reported error parameters, we are able to best match the classical results.

4.2. Ensemble principal components classifiers over CICIDS 2017

Model description We extend the model reported in the previous section. We propose a new way of computing summations reported in Equation 1. We compute y_i using both cosine similarity and correlation measure between $\boldsymbol{v}_i$ and $\boldsymbol{z}$, in addition to the dot product between the two vectors as in the basic formula. Therefore, we have three summations concerning the major components and three summations regarding the minor ones (one for each method used to compute y_i). So in this model, to classify a sample, we also add in OR to Equation 2 conditions relative to the summations computed using cosine similarity and correlation measure, both for major and minor components. For what concerns the quantum version of the model, it does not change so much from the quantum simulation of the previous one. Indeed, we have always to retrieve a classical description of $\boldsymbol{v}_i$ and λ_i, such that $\|\boldsymbol{v}_i - \overline{\boldsymbol{v}}_i\|_2 \leq \delta$ and $\|\lambda_i - \overline{\lambda}_i\|_2 \leq 2\epsilon\sqrt{\lambda_i}$ in time $\widetilde{O}\left(\frac{\|\boldsymbol{X}\|\mu(\boldsymbol{X})kd}{\theta\sqrt{p}\epsilon\delta^2}\right)$ and $\widetilde{O}\left(\frac{\|\boldsymbol{X}\|\mu(\boldsymbol{X})k}{\theta\sqrt{p}\epsilon}\right)$ respectively, given an input data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. Then, we are able to compute the summations using cosine similarity, correlation measures, and dot product between the estimated vectors.

Results In this case, we report the results over the CICIDS 2017 dataset always considering the PCA70-based model. We fit our model over a training set of 5000 normal samples and we measure our performance over a test set of 87,300 normal and 70,000 attack samples. The reader can check the full thesis for the comparison between classical and quantum results. Here we want to show the performance improvement of our ensemble model with respect to the model described in Section 4.1. We report only the quantum cases for easiness of visualization (error parameters $\epsilon_\theta = \epsilon = 1, \delta = 0.1$, and $\eta = 0.1$). By using the ensemble model there is a performance increase, more precisely in recall and accuracy. This was expected because the main characteristic of this new model is that, instead of two conditions in OR (as in Equation 2), we have six for classifying a sample as an attack. Having more conditions in OR increases the probability of classifying a sample as an attack, increasing the FP and reducing the FN cases. Indeed, for each α, we can also notice lower precision results for the ensemble model with respect to the precision of q-PCA70 with major and minor components. However, the increase in recall is much more evident than the decrease in precision, and therefore also the f1-

α	recall		precision	
	q_1	q_2	q_1	q_2
1%	0.3605	0.3980	0.9694	0.9530
2%	0.5897	0.7384	0.9654	0.9407
4%	0.6330	0.8961	0.9436	0.9079
6%	0.6337	0.9696	0.9161	0.8725
8%	0.6443	0.9756	0.8899	0.8345
10%	0.6590	0.9778	0.8692	0.8044

α	f1-score		accuracy	
	q_1	q_2	q_1	q_2
1%	0.5255	0.5615	0.7036	0.7169
2%	0.7322	0.8274	0.8035	0.8597
4%	**0.7577**	0.9019	0.8156	0.9113
6%	0.7492	**0.9185**	0.8068	0.9216
8%	0.7475	0.8995	0.8017	0.9008
10%	0.7497	0.8827	0.7996	0.8816

Table 2: Results comparison for q-PCA70 with both major and minor components (q_1) and q-PCA70 ensemble (q_2) over CICIDS 2017.

score increases.

4.3. PCA with reconstruction loss over CICIDS 2017

Model description We use this anomaly detection PCA-model following the work by Verkerken et al. [18]. In short, it works in the following way: given an input data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, we fit a PCA model specifying the number of principal components k to retain, with $k \leq d$. Then, we map the original data into the PCA feature space $\mathbb{R}^{n \times k}$. Finally, we re-project back the data into the original feature space. This process of mapping in a lower-dimensional space and reconstructing data inevitably brings to a reconstruction error. Therefore, we compute the loss as the sum of square error (SSE) between the original and transformed data and we use it as anomaly score. Then, using a threshold, we classify each sample based on its anomaly score. For what concerns the quantum counterpart, we have to retrieve a classical description of the top-k principal components with error δ in $\widetilde{O}\left(\frac{\|\boldsymbol{X}\|\mu(\boldsymbol{X})kd}{\theta\sqrt{p}\epsilon\delta^2}\right)$ steps to be able to perform the process of mapping and reconstructing data also for the quantum model.

Results In Table 3, we report the results obtained by executing this PCA-based model with 32 principal components over the CICIDS dataset, where we have a training set of 158,000 normal samples, and a test set of 50,000 normal plus 12,000 DDoS samples. In Table 4, we report different quantum results at the increase of the error δ. The other error parameters are $\epsilon_\theta = \epsilon = 0.3$, and $\eta = 0.00075$. With these error parameters, we are able to extract the same number of principal components of the classical model. It is important to get the same classical number of principal components to be consistent in the comparison between quantum and classical performances. Therefore, we keep those parameters fixed by varying δ instead, which we remember to be the error related to the estimation of singular vectors. We can notice that, by taking an error $\delta = 0.01$, we obtain the same classical performances, reported in Table 3. In general, we can see that by increasing δ, precision and accuracy decrease while recall increases.

Recall	Precision	F1-Score	Accuracy
0.9912	0.9128	0.9504	0.9808

Table 3: Results of PCA with reconstruction loss with 32 principal components (that retain 99.75% of the variance) over CICIDS 2017 normalized with QuantileTransformer() with 24 quantiles. The outlier threshold used is $t = 0.06632108379654125$.

δ	Recall	Precision	F1-Score	Accuracy
0.01	0.9912	0.9128	0.9504	0.9808
0.1	0.9917	0.9123	0.9503	0.9808
0.9	0.9979	0.7131	0.8318	0.9252
2	1.0	0.2730	0.4289	0.5066

Table 4: Results of q-PCA with reconstruction loss over CICIDS 2017 with different values of δ error.

5. Error analysis

In the experiments in Section 4.1 and 4.2, we only report cases of errors that we tolerate to obtain performance equal to the classical ones. If we increase both the error δ and ϵ, which are the errors that we use to estimate singular vectors and singular values respectively, we find that the performances decrease. In general, we have seen that ϵ error impacts more than δ in performances. The main reason is that the ϵ error is used to estimate singular values which also determine the number of singular vectors extracted by quantum procedures [2]. Therefore increasing ϵ could cause a wrong number of principal components retained by quantum models, leading to completely random predictions. We notice the same problem if we increase ϵ_θ and η in the quantum binary search routine. For what concerns the error δ, we have seen that thanks to the tomography property shown in Section 3.1, we can insert even a bigger δ error and still obtain good performances since the tomography returns estimates with a lower error than δ. We can notice this property in the results reported in Section 4.3. Indeed, we have shown that increasing δ from 0.01 to 0.1 does not change the intrusion detection capability of the model. Even if with $\delta = 0.9$ the performance does not decrease as we could expect with such a big error. By increasing δ up to 2 the performance decrease a lot. Therefore, this tomography property allows us to insert a bigger error, as 0.1 instead of 0.01, to estimate singular vectors, to obtain practically the same performance, and, at the same time, to impact less on the running time given that the time complexity of the tomography scales as $\sim O\left(\frac{d}{\delta^2}\right)$, where d is the vector length.

6. Running time comparison

Let us focus on the experiment reported in Section 4.1 about the principal components classifier model with only major components over KDDCUP. We report the running time comparison between the quantum and classical model's extraction. Once we find the errors that we can tolerate, we fix them in the running time formula and, by varying the number of samples and features, we see how the running time changes. In our experiments, we fix the probability of failure $\gamma = \frac{1}{d}$, with d number of features of the input data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. In this case, we compare the quantum running time with a randomized classical version of PCA, already implemented in Sklearn, which has a complexity of $O(nd \log k)$[7], with n, d number of samples and features respectively and k number of principal components retained. In Figure 2, we report this comparison with the blue and green planes representing the quantum and classical running time respectively. As expected, for small datasets, we do not have an advantage in using quantum machine learning in terms of running time. As the dimensionality of the dataset increases, the advantage of quantum machine learning becomes increasingly clearer. In the next section, we discuss the results found reporting some real examples.

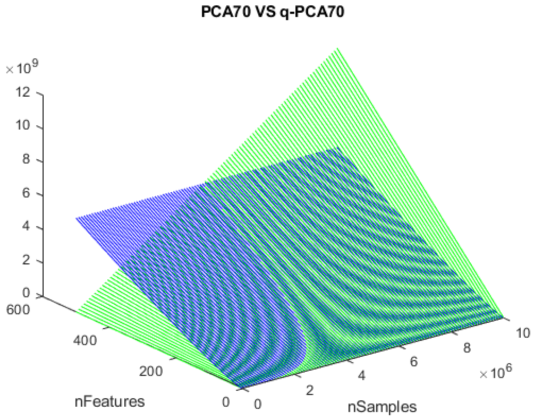

Figure 2: Running time comparison between quantum (blue) and classical (green) PCA70 with only major components model over KDDCUP.

7. Running time analysis

Figure 2 shows that with $\approx 4 * 10^6$ samples of ≈ 50 features, quantum machine learning starts to exhibit a speed-up over classical machine learning. These numbers of samples and features are reasonable for intrusion detection since they are in line with the number of samples and features of publicly available datasets, such as KDDCUP or NSL-KDD. However, for a dataset of $4 * 10^6$ samples and 50 features, the quantum speed-up is minimal: we need $\approx 3.5 * 10^8$ steps to fit the quantum model against $\approx 4.6 * 10^8$ for the classical one. By keeping fixed the number of features to 50 and by increasing the number of samples to 100, 000, 000, we start to see a great advantage in using quantum machine learning. Indeed, fitting the quantum model with a dataset of that size requires $\approx 2.3 * 10^7$ steps against $\approx 1.1 * 10^{10}$ of the approximated classical model. Even in this case, 100, 000, 000 samples are not a big number for intrusion detection tasks. Just think of the potential network connections or packets received by big companies such as Netflix or Amazon that count more than 200 million subscribers, considering that also non-subscribers can connect or send network packets to the servers of these companies.

Even the magnitude of the latest recent cyberattacks, in the order of millions of packets per second, reported by big companies such as Microsoft [17] and Amazon [1], or services such as GitHub [9] and WordPress [12] can make us conclude that, nowadays, 100,000,000 samples are realistic for network systems and so for intrusion detection tasks.

If we fix a large number of features such as 500, we can see that by increasing the number of samples, quantum is even more advantageous than classical machine learning. However, in intrusion detection tasks we do not deal with such high dimensional data. In this case, probably, malware detection could benefit from using quantum machine learning due to the increasing complexity of malware that may require lots of features to be described [3, 5, 13]. Future development in this direction could be interesting.

In the full thesis, we also reported the comparison between quantum and classical running time of the model whose results are reported in Table 4. In this running time comparison, it is even more clear the impact of the error on the running time. Moreover, we report a comparison between quantum and classical running time of the PCA-based model in which we need to extract both major and minor components. In this case, quantum machine learning is not so advantageous as classical machine learning in solving intrusion detection problems. The reader can check the full thesis for these results.

8. Conclusions

In this work, we theoretically analyze possible applications of quantum machine learning in network intrusion detection problems. One of the main contributions is the open-source framework we have developed to simplify the simulation process of quantum algorithms. We verify the correctness of this framework with tests over the main quantum routines bringing interesting insights into tomography. Using this framework, we study how the error of quantum machine learning algorithms affects the performances of quantum anomaly detection models. We also report a study of the running time by comparing quantum and classical machine learning anomaly detection models to derive advantages and disadvantages in their usage.

9. Future Works

This work aims to pave the way for other research on the application of quantum machine learning in cybersecurity. Therefore, we leave for future works to extend the framework that we started to develop by simulating new quantum machine learning algorithms to solve intrusion detection problems. We also propose to focus on other tasks instead of network intrusion detection, such as malware detection where, due to the increasing complexity of malware, we would need more and more features resulting in datasets with ever-larger dimensionality. An in-depth study of the probability of failure of quantum machine learning algorithms, as we did for the error, could be interesting to evaluate running times even more in detail.

References

[1] BBC. Amazon 'thwarts largest ever ddos cyber-attack', 2020. URL `https://www.bbc.com/news/technology-53093611`.

[2] A. Bellante, A. Luongo, and S. Zanero. Quantum algorithms for data representation and analysis, 2021. URL `https://arxiv.org/abs/2104.08987`.

[3] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu. Large-scale malware classification using random projections and neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3422–3426, 2013. doi: 10.1109/ICASSP.2013.6638293.

[4] J. Ding and B.-Y. Yang. *Multivariate Public Key Cryptography*, pages 193–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-88702-7. doi: 10.1007/978-3-540-88702-7_6. URL `https://doi.org/10.1007/978-3-540-88702-7_6`.

[5] D. Gibert, C. Mateu, and J. Planes. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, 03 2020. doi: 10.1016/j.jnca.2019.102526.

[6] A. Habibi Lashkari, G. Kaur, and A. Rahali. Didarknet: A contemporary approach

to detect and characterize the darknet traffic using deep image learning. In *2020 the 10th International Conference on Communication and Network Security*, ICCNS 2020, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450389037. doi: 10.1145/3442520.3442521. URL `https://doi.org/10.1145/3442520.3442521`.

[7] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev., Survey and Review section, Vol. 53, num. 2, pp. 217-288, June 2011*, 2009. doi: 10.48550/ARXIV.0909.4061. URL `https://arxiv.org/abs/0909.4061`.

[8] I. Kerenidis and A. Prakash. A quantum interior point method for lps and sdps, 2018. URL `https://arxiv.org/abs/1808.09266`.

[9] S. Kottler. Github ddos attack in 2018, 2018. URL `https://github.blog/2018-03-01-ddos-incident-report/`.

[10] D. Micciancio and O. Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-88702-7. doi: 10.1007/978-3-540-88702-7_5. URL `https://doi.org/10.1007/978-3-540-88702-7_5`.

[11] NIST. Getting ready for post-quantum cryptography: Exploring challenges associated with adopting and using post-quantum cryptographic algorithms, 2021. URL `https://csrc.nist.gov/publications/detail/white-paper/2021/04/28/getting-ready-for-post-quantum-cryptography/final`.

[12] L. Segall. Wordpress hammered by massive ddos attack, 2011. URL `https://money.cnn.com/2011/03/03/technology/wordpress_attack/index.htm`.

[13] A. Shabtai, Y. Fledel, and Y. Elovici. Automated static code analysis for classifying android applications using machine learning. In *2010 International Conference on Computational Intelligence and Security*, pages 329–333, 2010. doi: 10.1109/CIS.2010.77.

[14] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pages 108–116, 01 2018. doi: 10.5220/0006639801080116.

[15] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)*, 01 2003.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani. A detailed analysis of the kdd cup 99 data set. *IEEE Symposium. Computational Intelligence for Security and Defense Applications, CISDA*, 2, 07 2009. doi: 10.1109/CISDA.2009.5356528.

[17] A. Toh. Microsoft ddos attack in 2021, 2022. URL `https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends`.

[18] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck. Unsupervised machine learning techniques for network intrusion detection on modern data. In *2020 4th Cyber Security in Networking Conference (CSNet)*, pages 1–8, 2020. doi: 10.1109/CSNet50428.2020.9265461.