Quantity, Quality, and Quantum Perspectives in Data-Driven Cybersecurity

Mansur Ziiatdinov[1,*,†], Salvatore Distefano[1,†]

[1]University of Messina, Italy

Abstract

This paper addresses the critical challenge of imbalanced datasets in data-driven cybersecurity, affecting model efficacy. It investigates the impact of data *quantity* and *quality* management through random undersampling on multiclass classification using the UNSW-NB15 dataset and Support Vector Classifiers (SVC). Furthermore, it explores a *quantum perspective* by examining quantum SVC with different sampling methods. Obtained results show that balanced datasets, achieved via undersampling, yield superior classification performance. While quantum approaches demonstrate fast learning, current noise limitations are noted. This work underscores the importance of data preprocessing and highlights future avenues for quantum machine learning in robust threat detection.

Keywords

Cybersecurity Analytics, Network Intrusion Detection, Data Quality, Data Balancing, Undersampling, Machine Learning, Quantum Computing, Quantum Machine Learning, Multiclass Classification, UNSW-NB15 Dataset.

1. Introduction

Cybersecurity is a fundamental quality aspect of Information and Communication Technologies aiming to, e.g., restrict access to sensitive data, protect critical infrastructure from disruption, and ensure national security against digital espionage and sabotage. The digital landscape poses persistent and increasingly difficult threats, challenges, and evolving issues, including sophisticated ransomware attacks, widespread phishing campaigns, distributed denial-of-service (DDoS) attacks, and insider threats, to name a few. Such problems may be affected by rapidly evolving attack vectors, the increasing complexity of systems, human error, and the growing vulnerabilities in Internet of Things (IoT) devices. To address them, a combination of proactive and reactive solutions is consistently required. Proactive measures include, e.g., threat intelligence, vulnerability management, security by design principles, encryption, firewalls, and intrusion prevention systems. Reactive measures encompass approaches such as robust incident response frameworks, digital forensics, prompt patch management, intrusion detection systems (IDS), and comprehensive malware analysis.

Models are of strategic importance in addressing both approaches. Models are usually used to design and to assess computing system Cybersecurity aspects, and are particularly useful for predicting and identifying threats as well as for assessing their impact. But, for such a purpose, they need large amount of data to either fit statistical parameters and distributions of white-box models or to train black-box (e.g. machine learning) ones. This is moving the focus of cybersecurity research on datasets and data management, towards *data-driven cybersecurity*.

As a consequence, cybersecurity data management presents significant challenges regarding data availability, quantity and size, i.e. the number of data points available in a dataset. Cybersecurity datasets, indeed, are typically quite unbalanced. Attack events, such as man-in-the-middle attacks, worm infections, malware propagation, denial-of-service (DoS) attacks, SQL injection, cross-site scripting (XSS), and privilege escalation, are inherently rare in real-world observations. Furthermore, cybersecurity data

QualITA 2025: The Fourth Conference on System and Service Quality, June 25 and 27, 2025, Catania, Italy

*Corresponding author.

†These authors contributed equally.

mansur.ziiatdinov@unime.it (M. Ziiatdinov); salvatore.distefano@unime.it (S. Distefano)

0000-0001-7415-2726 (M. Ziiatdinov); 0000-0003-3448-961X (S. Distefano)

quality impacts on quantity, implying preprocessing activities such as filtering, cleaning, aggregation, and fusion that may affect the dataset size. Data scarcity in cybersecurity may lead to biased and misleading representation of normal versus anomalous activities.

Considering classification problems and machine learning models, therefore, dealing with data quantity and quality in cybersecurity analysis is of utmost importance. Widely unbalanced datasets may lead to model overfitting, where the model learns the majority class too well and performs poorly on the rare attack instances. Conversely, small datasets can result in low accuracy and generalizability, failing to capture the complexity of real-world phenomena.

This paper performs an in-depth analysis of unbalanced datasets. It applies undersampling techniques to obtain balanced datasets with different sizes. These modified datasets are then compared against their original, unbalanced counterparts. The objective is to determine a proper size and optimal mixture among classes for effective classification problems. Even different computational paradigms are considered within this comparison, including classical and quantum computing approaches.

The results obtained from investigating a real-world cybersecurity dataset provide interesting insights. The study demonstrates the effectiveness of random undersampling techniques. Furthermore, it reveals better performance with balanced datasets, as opposed to unbalanced ones. This trend is much more evident and further highlighted by increasing the dataset size to widely unbalanced ones.

The remainder of the paper is organized as follows. Section 2 introduces and defines data-driven cybersecurity, while Section 3 provides the background solutions and technologies for its enforcement. Section 4 reports a case study on network intrusion detection systems based on the well-known UNSW-NB15 dataset [1, 2], dealing with classification problems exploiting both classical and quantum ML models. Section 5 closes the paper with some final remarks.

2. Data-driven Cybersecurity

Data-driven cybersecurity is an emerging approach that leverages data science and technologies such as advanced analytics, machine learning, and vast amounts of security-related data to enhance the detection, prevention, and response to cyber threats [3]. Data-driven security fundamentally redefines the approach to cyber defense by leveraging computational analysis of extensive datasets to detect, predict, and mitigate threats. This paradigm shifts from reactive, signature-based detection to a proactive, adaptive posture, driven by insights extracted from various security telemetry sources [4]. The core principle involves transforming raw, heterogeneous security data—including network traffic logs, system event records, user behavior profiles, and vulnerability scans—into actionable intelligence through sophisticated analytical techniques.

This paradigm shift from traditional, reactive security measures to proactive data-centric methodologies has become essential as cyber threats grow in complexity and frequency. The increasing complexity and frequency of cyber threats necessitate this evolution. By processing structured, semi-structured, and unstructured data, organizations can derive actionable insights that inform robust security strategies, thereby enhancing overall resilience against cyber attacks [5].

At its heart, data-driven security relies heavily on machine learning (ML) and deep learning algorithms. These models are trained on historical and real-time data to identify intricate patterns indicative of malicious activity or anomalies that deviate from established baselines [6]. This approach harnesses advanced analytics, machine learning (ML), and extensive security-related datasets to improve the detection, prevention, and response to cyber threats [4]. ML algorithms facilitate real-time adaptation and anomaly detection, significantly reducing the time required to identify and mitigate threats. Empirical evidence suggests that security teams employing data-driven techniques exhibit a swifter response to security breaches compared to those relying on conventional methods, underscoring the criticality of data-centric strategies in modern cybersecurity frameworks [7]. Furthermore, the application of predictive analytics empowers organizations to anticipate potential threats before their materialization, marking a significant evolution in the cybersecurity landscape [8].

Despite its demonstrable advantages, data-driven cybersecurity encounters several inherent chal-

lenges. These include issues pertaining to data quality, the complexities of regulatory compliance, and the imperative for effective inter-team collaboration. The reliance on diverse data sources necessitates rigorous data management practices to ensure the accuracy and relevance of information, as suboptimal data quality can inherently compromise threat detection capabilities. Moreover, organizations must diligently navigate the intricate landscape of evolving data privacy regulations while simultaneously optimizing model performance to mitigate false positives and combat alert fatigue. Data-driven security systems, particularly in areas like smart grids, face challenges in integrating diverse data sources and ensuring real-time analysis [9]. Similarly, for Android systems, flexible data-driven security models are explored to address emerging threats [10]. Smart cities also present opportunities and challenges for data-driven cybersecurity [11].

As the cybersecurity landscape continues its dynamic evolution, the emphasis on data-driven methodologies is projected to intensify, solidifying its role as a vital component of organizational security strategies. The continuous progression of technologies such as artificial intelligence (AI) and machine learning is poised to further amplify the potential of data-driven cybersecurity in advanced threat management, thereby reinforcing its indispensable significance in safeguarding sensitive information within an increasingly digitized global environment. This includes applications in industrial control systems, where multilayer data-driven cyber-attack detection systems are being developed based on network, system, and process data [12]. The big data era also emphasizes the shift from merely securing big data to leveraging data for enhanced security [13]. This data-driven approach aims to improve autonomous systems through data analytics and cybersecurity [14].

3. Methods

3.1. Machine learning

Machine learning (ML) enables systems to learn patterns from data without explicit programming. It involves training models on datasets to make predictions or decisions. This process involves defining a problem, collecting a dataset, data cleaning and feature engineering, selecting and training a model, evaluating it and deploying the trained model. This section contains brief introduction to the ML; for more detailed exposition please refer, for example, to [15].

A dataset is a structured collection of data items, typically represented as a matrix where rows correspond to samples and columns represent features. For the supervised learning, the dataset additionally contains the labels for each data item that represent the ground truth value. Formally, the dataset is denoted by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{m \times n}$, the corresponding labels are denoted by $y_i \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary finite set, $i = 1, \ldots, n$, and each class is denoted by $\mathbf{C}_k = \{\mathbf{x}_i \in \mathbf{X} \mid y_i = k\}$, where $k \in \mathcal{C}$. By definition, $\mathbf{C}_j \cap \mathbf{C}_k = \emptyset$ for $j \neq k$, and $\bigcup_{k \in \mathcal{C}} \mathbf{C}_k = \mathbf{X}$. Once the dataset is defined, the next step is to split it into training and testing subsets to evaluate model performance.

The dataset $\mathbf{X}$ is divided into training $\mathbf{X}_{\text{train}}$ and testing $\mathbf{X}_{\text{test}}$ subsets to evaluate model performance: $\mathbf{X} = \mathbf{X}_{\text{train}} \cup \mathbf{X}_{\text{test}}$; $\mathbf{X}_{\text{train}} \cap \mathbf{X}_{\text{test}} = \emptyset$. The training set is used to train the model, while the test set assesses its generalization to unseen data. Common splits include 80% training and 20% testing, with stratified sampling to maintain tasks (i.e. the training and testing datasets contain the same proportion of items of different classes). This ensures that the model performance is measured on representative data.

Building on this, multi-class classification is a supervised learning task where the goal is to assign input data to one of multiple classes. Unlike binary classification, which distinguishes between two classes, multi-class problems require models to differentiate between three or more categories (e.g., classifying images into "cat," "dog," "bird", or, as in our case, different types of attacks). To evaluate such models, specific metrics are used.

One of the metrics to evaluate the model effectiveness is the F_1 score. This metric balances precision ($\text{TP}/(\text{TP} + \text{FP})$) and recall ($\text{TP}/(\text{TP} + \text{FN})$), addressing trade-offs between false positives (FP) and false negatives (FN). For multi-class problems, the macro-averaged F_1 score, the unweighted mean of F_1 score for all classes, provides a balanced measure of model performance.

There are a multitude of methods that can be used to solve the classification problem. One such method is the use of support vector machines (SVMs). SVMs identify the optimal hyperplane to separate classes in feature space. The Support Vector Classifier (SVC) is an implementation of SVM for classification. Formally, SVC solves the following optimization problem

$$\begin{aligned} \underset{(\alpha_1,\ldots,\alpha_m)}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{m} \alpha_j - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j), \\ \text{s.t.} \quad & 0 \leq \alpha_j \leq C \text{ and } \textstyle\sum_{i=1}^{m} \alpha_i y_i = 0, \end{aligned}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ denotes the dot product, the weights α_j are associated with the support vectors (the points closest to the separating hyperplane), and C is a regularization hyperparameter.

A key technique in SVMs is the kernel trick, which maps input data into a higher-dimensional space using an arbitrary kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ instead of the dot product. This allows linear separation of non-linearly related data without explicitly computing the transformation, improving efficiency and flexibility. One of the most widely used kernels is the Radial Basis Function (RBF) kernel.

The RBF kernel, defined as $K(x, y) = \exp(-\gamma \|x - y\|^2)$, where γ controls the kernel influence, is a popular choice for non-linear problems. By mapping data into an infinite-dimensional space, it effectively captures complex, non-linear relationships. This makes the RBF kernel particularly valuable in scenarios where data is not linearly separable.

3.2. Imbalanced datasets and sampling

In machine learning, sampling refers to the process of selecting a subset of data from a larger dataset to train models. This is particularly critical when dealing with imbalanced datasets, where one class (the majority class) significantly outnumbers the other (the minority class): i.e. if some class $\mathcal{C}_i$ outnumbers other classes $\mathcal{C}_j$: $|\mathcal{C}_i| \gg |\mathcal{C}_j|$. Proper sampling techniques help ensure that models are trained on representative data, reducing bias and improving generalization. Building on this, undersampling methods are specifically designed to address class imbalance by removing some instances in the majority class.

To address class imbalance, undersampling techniques are employed to balance the dataset by decreasing the number of majority class samples. Two common approaches are:

Random undersampling. This method randomly selects a subset of the majority class samples to match the number of minority class samples. For example, if the minority class has 100 instances and the majority class has 1,000, random undersampling would randomly retain majority samples. While simple and computationally efficient, this approach risks losing important information from the majority class, as random selection does not consider the distribution or relevance of the samples.

NearMiss methods. There are several closely related undersampling techniques that selects majority class samples based on their proximity to the minority class. This study focuses on NearMiss-3 method, which operates in three steps:

- For each minority sample, identify its k nearest neighbors from the majority class.
- select the k samples that are farthest from the minority sample (to avoid overfitting to noisy or outlier samples).
- Retain these selected majority samples for the balanced dataset.

This method preserves the structure of the majority class while focusing on regions where the minority class is most prevalent, making it more effective than random undersampling in retaining meaningful patterns.

By incorporating these undersampling strategies, it is possible to mitigate the effects of class imbalance and achieve robust and fair model performance.

3.3. Quantum computing

Quantum computing leverages principles of quantum mechanics, such as superposition and entanglement, to perform computations that are infeasible for classical computers. Unlike classical bits, which exist in a state of 0 or 1, quantum systems use qubits (quantum bits) to encode information in a superposition of states. This enables parallelism and exponential speedups for specific problems, such as factoring large numbers or simulating quantum systems. This section provides a basic introduction into quantum computing; for a more detailed introduction please refer, for example, to the textbook [16].

A qubit is the fundamental unit of quantum information. It is represented as a quantum state in a two-dimensional Hilbert space:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ are complex amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$. The basis states $|0\rangle$ and $|1\rangle$ correspond to classical bits, but the qubit can exist in any linear combination of these states.

For n qubits, the combined quantum state lives in a 2^n-dimensional Hilbert space. For example, two qubits can be in a state:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + 10\rangle + \delta|11\rangle,$$

where $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. This state can exhibit entanglement, where the qubits' states are correlated in ways impossible for classical systems. A classic example is the Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

The notation $|j\rangle$ for a number $j \in \mathbb{Z}_{\geq 0}$ means that $|j\rangle$ is a basis state with qubit states corresponding to the binary notation of the number j. For example, the state $|10\rangle$ can be written as $|2\rangle$, and in general, the state of an n-qubit system is written as $|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle$.

Quantum states can be modified by performing quantum operations. They are described by unitary matrices U, which preserve the norm of the quantum state. For a single qubit, the Hadamard gate H and Pauli gates X, Y, Z are fundamental:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Pauli gates can be used to define rotation gates, for example, $R_Y(\theta) = e^{-iY\theta/2}$, where $\theta \in [0, 2\pi)$:

$$R_Y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}.$$

Another important gate is the controlled-NOT (CNOT) gate, which is analogous to the "if-then-else" construction in the classical programming:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

To extract the information from the quantum state, one needs to perform measurements and get a classical outcome. For a qubit in state $|\psi\rangle = \sum_{j=0}^{2^n-1} \alpha_j |j\rangle$, the probability of measuring $|j\rangle$ is $|\alpha_j|^2$. After the measurement, the state collapses to the observed basis state, so subsequent measurements will return the same outcome with probability 1.

Quantum Machine Learning (QML) combines principles from quantum computing and machine learning. It is used to process and analyze data, often offering advantages in speed, scalability, or expressivity over classical methods. QML typically outsources some step in the ML pipeline to a quantum computer. In order to work with classical data on a quantum computer, it is necessary to encode it in a quantum state.

Data encoding is the process of translating classical input data into quantum states. Common encoding strategies include:

- amplitude encoding that represents 2^q-dimensional data as the amplitudes of a q-qubit quantum state:

$$|\psi(\mathbf{x}_i)\rangle = \frac{1}{\|\mathbf{x}_i\|} \sum_{j=1}^{m} x_{i,j} |j\rangle ,$$

- angle encoding that represents data as the angles of rotations, for example:

$$|\psi(\mathbf{x}_i)\rangle = \prod_{j=1}^{m} R_Y^{(j)}(x_{i,j}) \left|0^{\otimes m}\right\rangle ,$$

The choice of encoding depends on the task, the quantum hardware, and the desired trade-off between expressivity and complexity.

This study adopts a heuristic `real_amplitudes` encoding $\left|\psi_{\text{qra}}(\mathbf{x}_i)\right\rangle$ from the Qiskit circuit library [17], denoted here as QRA. This technique involves several layers of angle encoding interspersed with entanglement layers. Since it is difficult to express the encoded state in closed form, the corresponding circuit is described. Let the number q of qubits and the number of layers L be such that $n = qL$, where n is the number of features. Then

$$\left|\psi_{\text{qra}}(\mathbf{x}_i)\right\rangle = U_1(\mathbf{x}_i) V U_2(\mathbf{x}_i) V \times \cdots \times V U_L(\mathbf{x}_i),$$

where $U_j(\mathbf{x}_i) = \bigotimes_{k=1}^{q} R_{\text{Y}}(x_{i,jq+k})$ rotates the qubits to encode some of the features (i.e. $x_{i,(j-1)q+1}, x_{i,(j-1)q+2}, \ldots, x_{i,(j-1)q+q}$). The entanglement layer V performs CNOT gates between i-th and $i+1$-th qubits for $i \in \{n-2, n-3, \ldots, 1, 0\}$. See Figure 1 for an example.

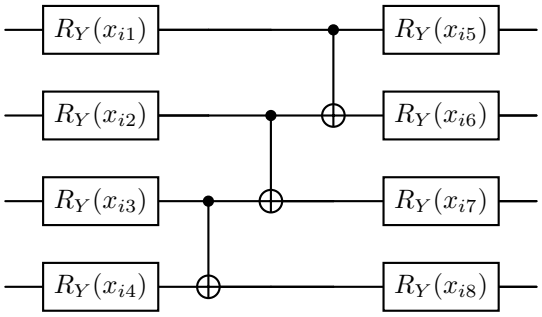

Figure 1: RA circuit for $n = 8, q = 4, L = 2$

4. Case study and experiments

4.1. Dataset and testbed

This study is based on the widely adopted UNSW-NB15 dataset [1, 2] created for network intrusion detection systems. Additionally, it has been cleaned from contaminant features [18]. Each data item in the dataset is the description of a network packet. After the cleaning and encoding the categorical feature as an ordinal, the training dataset contains $n = 175\,341$ data items with $m = 32$ numeric features, while the test dataset contains $82\,332$ data items with the same features. Each data item is associated with two labels: first, it is labeled as a "normal" packet or as an "attack" packet; second, it is further labeled by different attack types, i.e. "Backdoor", "DoS", "Exploits", etc. (10 classes in total). The dataset is highly imbalanced: for example, the smallest class "Worms" contains only 130 data items, while the "Exploits" class contains $33\,393$ data items. The whole dataset is denoted by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{m \times n}$, the corresponding labels are denoted by $y_i \in \mathcal{C}$, where $\mathcal{C} = \{\text{Backdoor}, \ldots, \text{Worms}\}$ and $i = 1, \ldots, n$, and each class is denoted by $\mathbf{C}_k = \{\mathbf{x}_i \in \mathbf{X} \mid y_i = k\}$, where $k \in \mathcal{C}$. By definition, $\mathbf{C}_j \cap \mathbf{C}_k = \emptyset$ for $j \neq k$, and $\bigcup_{k \in \mathcal{C}} \mathbf{C}_k = \mathbf{X}$.

All tests were performed on a machine with the following characteristics. CPU: AMD Ryzen 9 5950X, RAM: 64 GiB, OS: Linux, kernel: 6.6.74-gentoo, Python: 3.12.9, jupyter-core: 5.7.2, numpy: 2.2.2, qiskit: 1.3.2, qiskit-machine-learning: 0.8.2, scikit-learn: 1.6.1. The source code is available upon request.

4.2. Undersampling to Balance the Dataset

The case study focus on the effects of balancing the dataset on the multiclass classification of a packet into various attack classes. In this experiment, given the required class size ℓ, the random undersampling method selects a random subset $\mathbf{C}'_k \subseteq \mathbf{C}_k$ for each class $k \in \mathcal{C}$. If $\ell \leq |\mathbf{C}_k|$ then $|\mathbf{C}'_k| = \ell$; otherwise, the random undersampling method selects the whole class: $\mathbf{C}'_k = \mathbf{C}_k$, and $|\mathbf{C}'_k| < \ell$. After the random undersampling step, the support vector classifier [19] is trained to classify the data items on the selected subset $\mathbf{X}' = \bigcup_{k \in \mathcal{C}} \mathbf{C}'_k$ of training dataset and its performance is assessed on randomly selected 100 data items (10 data items per class) from test dataset by using F_1 score averaged over all classes.

The experiment results are reported in Figure 2. Note that the maximum average F_1 score roughly coincides with $\ell = 130$, which is the maximum ℓ value when the selected subset is still balanced, so training on an imbalanced dataset doesn't improve the results.

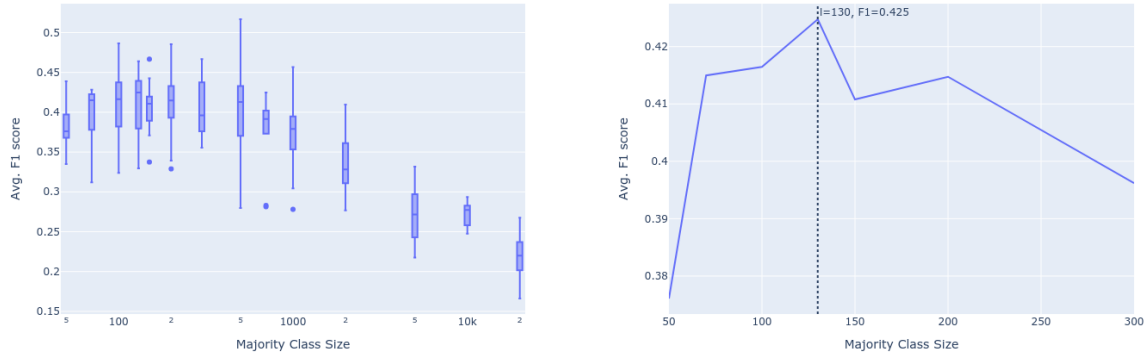

Figure 2: Average F1 score (higher is better) for different training set sizes: the left shows the distribution of values with boxplots (note the log-scale for x-axis); the right shows the median values in the region around the maximum value.

4.3. Quantum Approaches for Learning

Another (preliminary) result of this study is the investigation of the behavior of the quantum SVC in conjunction with different sampling techniques such as random undersampling and the NearMiss-3 method [20], which is based on geometric ideas. The heuristic quantum kernel `real_amplitudes` [17] (denoted here as QRA-8-4; the kernel uses 8 qubits and 4 layers of encoding gates) is compared with classical RBF kernel. In the experiment, the sampler selects a subset of the training set, and the SVC is trained on this subset. The average F_1 scores obtained in the experiment are reported on Figure 3. As it can be seen, the classical methods outperform the quantum one, and the random undersampling technique overperforms the NearMiss-3 method in both classical and quantum paradigms.

5. Conclusions

This study rigorously investigated the interconnected challenges of data *quantity* and *quality* in multiclass classification within data-driven cybersecurity, while also providing a preliminary *quantum perspective*. Our experiments with the UNSW-NB15 dataset unequivocally demonstrate the critical influence of data imbalance on model performance. We found that strategically managing data quantity

Figure 3: Average F1 score (higher is better) for quantum and classical training methods.

through random undersampling techniques to create balanced training sets significantly enhances classification accuracy and generalizability compared to using highly imbalanced datasets, thereby proving the direct link between data quality improvements and model efficacy. This finding highlights the essential role of diligent data preprocessing and balancing strategies in developing effective threat detection systems.

Furthermore, our preliminary exploration into the *quantum perspective* of machine learning, specifically with the quantum Support Vector Classifier, revealed promising fast learning capabilities. This suggests a compelling potential for quantum computing to offer novel computational paradigms for cybersecurity analytics in the future. However, it is crucial to acknowledge the current technological limitations, particularly concerning noise effects in contemporary quantum hardware, which underscore the need for continuous advancements in both quantum hardware and algorithm development.

Future work should focus on a deeper exploration of advanced data balancing techniques, including more sophisticated methods for managing data quantity and quality beyond simple undersampling. Investigating the robustness and scalability of quantum machine learning models on larger and more diverse cybersecurity datasets will be paramount. Additionally, developing and implementing effective noise mitigation strategies will be key to unlocking the full potential of quantum approaches and transitioning these theoretical perspectives into practical, high-impact cybersecurity applications.

Acknowledgments

We acknowledge financial support under the National Recovery and Resilience Plan (PNRR), Mission 4, Component 2, Investment 1.4, Call for tender No. 1031 published on 17/06/2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU, Project Title "National Centre for HPC, Big Data and Quantum Computing (HPC)" – Code National

Center CN00000013 – CUP D43C22001240001. This work has also been supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, project 3D-SEECSDE, CUP J33C22002810001, partnership on "SEcurity and RIghts in the CyberSpace" (PE00000014 - program "SERICS").

Declaration on Generative AI

During the preparation of this work, the author(s) used LLM tools in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. doi:10.1109/MilCIS.2015.7348942.

[2] N. Moustafa, J. Slay, Unsw-nb15, 2024. doi:10.34740/KAGGLE/DSV/9350725.

[3] M. Mattei, Data-Driven Cybersecurity: Reducing risk with proven metrics, Manning Publications, Shelter Island, NY, 2025.

[4] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, A. Ng, Cybersecurity data science: an overview from machine learning perspective, Journal of Big data 7 (2020) 1–29.

[5] J. Jacobs, B. Rudis, Data-driven security: analysis, visualization and dashboards, John Wiley & Sons, 2014.

[6] M. Hesham, M. Essam, M. Bahaa, A. Mohamed, M. Gomaa, M. Hany, W. Elsersy, Evaluating predictive models in cybersecurity: A comparative analysis of machine and deep learning techniques for threat detection, in: 2024 Intelligent Methods, Systems, and Applications (IMSA), 2024, pp. 33–38. doi:10.1109/IMSA61967.2024.10652833.

[7] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, Y. Xiang, Data-driven cyber security in perspective—intelligent traffic analysis, IEEE transactions on cybernetics 50 (2019) 3081–3093.

[8] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, Y. Xiang, Data-driven cybersecurity incident prediction: A survey, IEEE communications surveys & tutorials 21 (2018) 1744–1772.

[9] S. Tan, D. De, W.-Z. Song, J. Yang, S. K. Das, Survey of security advances in smart grid: A data driven approach, IEEE Communications Surveys & Tutorials 19 (2016) 397–422.

[10] D. Feth, A. Pretschner, Flexible data-driven security for android, in: 2012 IEEE Sixth International Conference on Software Security and Reliability, IEEE, 2012, pp. 41–50.

[11] N. Mohamed, J. Al-Jaroodi, I. Jawhar, Opportunities and challenges of data-driven cybersecurity for smart cities, in: 2020 IEEE systems security symposium (SSS), IEEE, 2020, pp. 1–7.

[12] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, J. Coble, Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data, IEEE Transactions on Industrial Informatics 15 (2019) 4362–4369.

[13] D. B. Rawat, R. Doku, M. Garuba, Cybersecurity in big data era: From securing big data to data-driven security, IEEE Transactions on Services Computing 14 (2019) 2055–2072.

[14] I. R. Noman, J. C. Bortty, K. K. Bishnu, M. M. Aziz, M. R. Islam, Data-driven security: Improving autonomous systems through data analytics and cybersecurity, Journal of Computer Science and Technology Studies 4 (2022) 182–190.

[15] S. J. Russell, P. Norvig, Artificial intelligence: A modern approach, Pearson, 2020. 4th edition.

[16] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, 2004. doi:10.1017/CBO9781107415324.004. arXiv:arXiv:1011.1669v3.

[17] M. Treinish, et al., Qiskit/qiskit: Qiskit 1.4.3, 2025. doi:10.5281/zenodo.15374661.

[18] L. D'Hooge, M. Verkerken, T. Wauters, B. Volckaert, F. De Turck, Discovering non-metadata

contaminant features in intrusion detection datasets, in: 2022 19th Annual International Conference on Privacy, Security & Trust (PST), 2022, pp. 1–11. doi:10.1109/PST55820.2022.9851974.

[19] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297. doi:10.1007/bf00994018.

[20] I. Mani, I. Zhang, knn approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of workshop on learning from imbalanced datasets, volume 126, ICML United States, 2003, pp. 1–7.