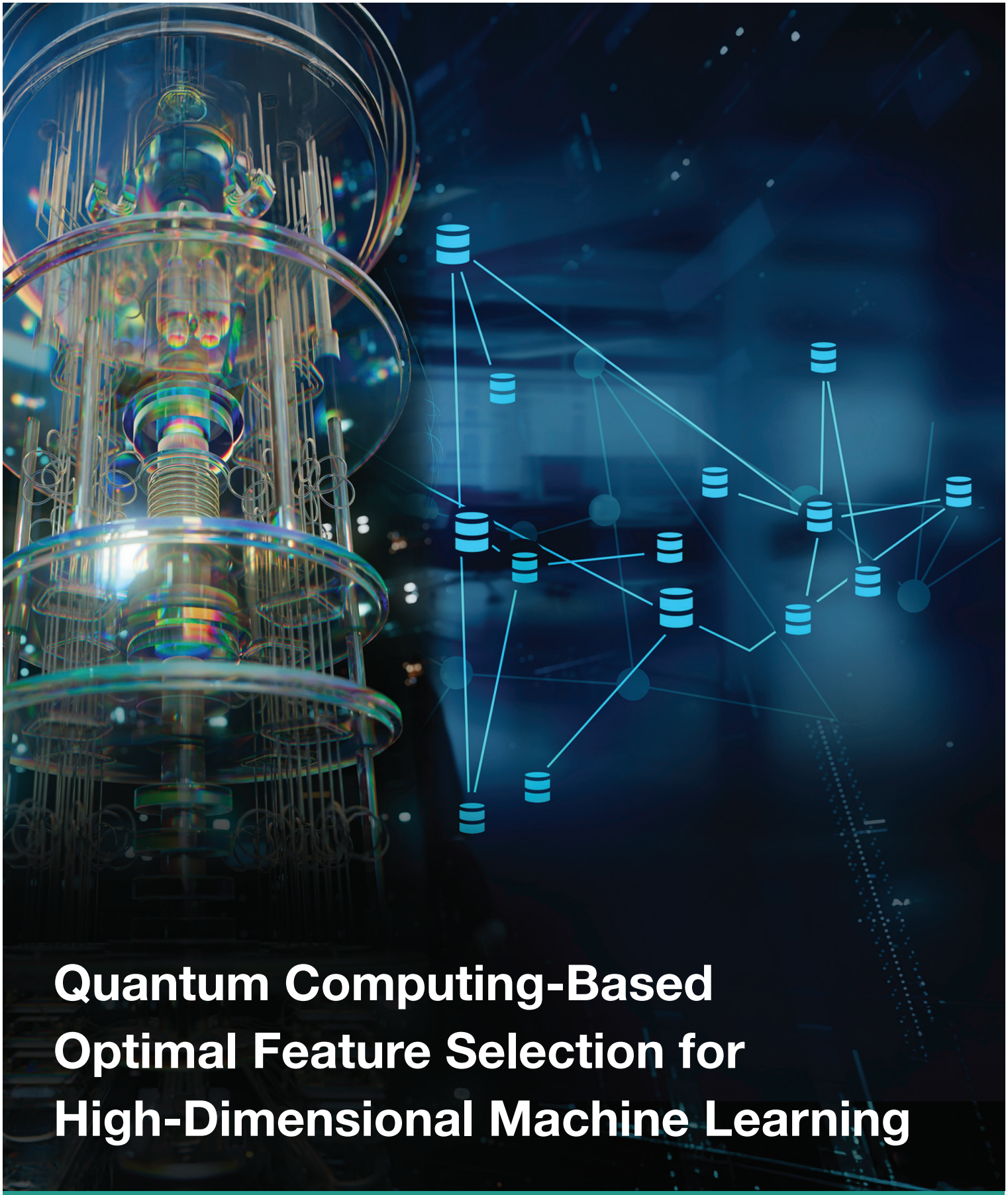


Quantum Computing-Based Optimal Feature Selection for High-Dimensional Machine Learning

Whitepaper by

Rohit Kumar Patel, AVP & Lead – Data Science and Quantum Computing, Mphasis

Dr. Revendranath T, Project Manager – Mphasis NEXT Labs | Advisor

Mphasis

The Next Applied

Contents

1. Executive Summary

Over the last two decades, Machine Learning (ML) has made great progress – leading to wider adaptation of Artificial Intelligence (AI) and transforming in businesses. Machine Learning works on the idea of learning from data generated by real-world processes. The learnings gained from the data is represented as a mathematical model.

The process of learning establishes a relationship between predictors and predicted variables. E.g., How income can function as a predictor variable to predict your ability to pay a loan. The predictor variables, known as features, used in training a model directly influence the performance of the trained model. Identifying and defining the right predictor variables as inputs to the ML model has been a challenge to the model developers especially in cases where either the number of features tends to become exceptionally large or relationships among features are complicated.

Using irrelevant or partially-relevant features in training an ML model negatively impacts the performance of the model and may lead to overfitting whereas excluding key features may result in underfitting. Therefore, careful selection and engineering of features play a crucial role in improving model KPIs such as accuracy, precision and recall where improving their generalization ability.

The process of feature selection requires domain and technical expertise to identify the relevant features without much loss of information. Many effective feature selection algorithms are available which assist the model developers in this process.

In this paper, we introduce an innovative approach to feature selection that harnesses the power of quantum computing. We use high-dimensional datasets in the domains of malicious software detection and Parkinson's disease detection to illustrate the application of the proposed feature selection methods. We additionally compare and benchmark our proposed method against various mainstream classical feature selection techniques.

2. Introduction/Business Context

The following graphic demonstrates the end-to-end Machine Learning development and deployment pipeline. Developing reliable predictive Machine Learning models in real-world situations requires careful selection of data, features, models and training techniques. Effective feature selection methods are integral to this pipeline as they significantly improve the training and prediction performance of ML methods.

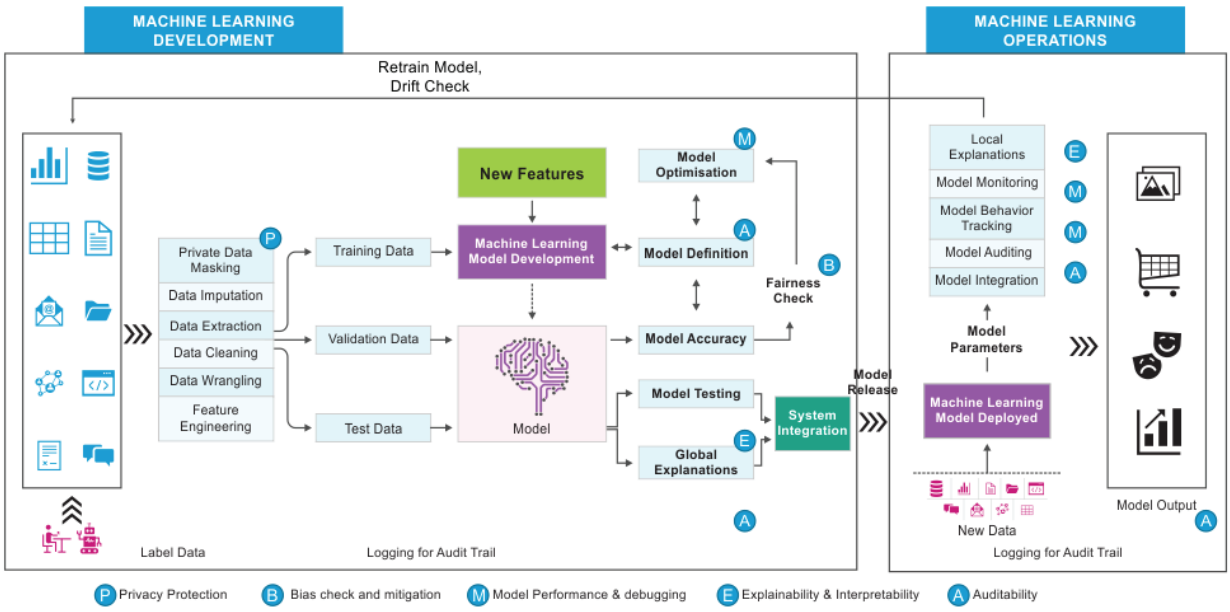


Figure 1: End-to-end Machine Learning development and deployment using MLOps principles

2.1 What is Feature Selection?

Feature selection is a critical aspect of feature engineering, which involves identifying and utilizing the most relevant data attributes to train Machine Learning models effectively. The feature selection process aims to reduce the number of input variables by eliminating redundant or irrelevant features, leaving only the most influential ones for the specific Machine Learning task. Implementing appropriate feature selection techniques not only streamlines the model training process by focusing on the most predictive features but also impacts training speed, enhances model performance and improves generalization performance, i.e., the ability to respond to unseen scenarios during the training process effectively.

Raw data serves as the fundamental input for Machine Learning algorithms. However, the high dimensionality of raw datasets can pose significant challenges, including excessive memory requirements, computationally intensive training processes and degraded model generalization performance due to the "curse of dimensionality" issue. Mathematically, a dataset can be represented as a matrix, where the columns correspond to features and the rows represent recorded samples. "Narrower" matrices that closely approximate the original data can address the "curse of dimensionality" problem. The process of finding these narrower matrices is known as dimensionality reduction, and it can be achieved through two main approaches: Feature extraction and feature selection.

Feature extraction involves deriving new data attributes from the existing ones, with the goal of preserving the most informative characteristics. In contrast, feature selection focuses on identifying and utilizing only the most relevant existing features, while disregarding those that do not significantly contribute to the learning process. Both techniques aim to enhance the performance of Machine Learning models. However, feature selection is preferred in scenarios where maintaining the original problem representation is critical or when the cost of acquiring and managing features is substantial.

2.1.1 Challenges in Feature Selection Process

Feature selection is an invaluable tool for data scientists, as it significantly enhances the efficacy of Machine Learning algorithms. The capability to identify and select pertinent features is crucial, as irrelevant, redundant or noisy features can impede an algorithm's performance, leading to diminished accuracy, increased computational costs and suboptimal learning outcomes.

Feature selection techniques must address the following four important challenges:

- **Class Imbalance**

Class imbalance refers to a scenario where the distribution of classes within a dataset is uneven, with one or more classes being significantly underrepresented compared to others. Class imbalances frequently occur in real-world applications such as credit card fraud, machine failure, network intrusions in cybersecurity, etc. In the context of feature selection, class imbalance can lead to biased models that disproportionately prioritize the features associated with the majority class. Addressing this issue is critical to ensure that the feature selection process does not unduly favor majority class features while overlooking potentially crucial features pertaining to minority classes.

- **Dataset Shift**

Dataset shift refers to changes in the underlying data distribution over time or across different operational environments. This phenomenon can pose challenges for feature selection methods, as the relationships between features and target variables may evolve, rendering the previously-selected features less relevant for new data distributions. Adapting feature selection techniques to effectively manage dataset shifts is crucial for maintaining robust model performance across varying scenarios and ensuring the continued relevance of the selected features.

- **Incremental Learning**

Incremental learning refers to the process of updating a model's knowledge base as new data becomes available over time. In the context of feature selection, this poses a challenge as the relative importance of features may shift with the introduction of new data patterns. To address this, incremental feature selection methods are necessary to adapt the model to evolving data trends, ensuring that the selected features remain relevant and informative as the model continues to learn from additional information.

- **Noisy Data**

Noisy data refers to datasets containing errors, outliers or irrelevant information. Feature selection methods may be susceptible to noisy data, as irrelevant or erroneous features may be incorrectly deemed important. Robust feature selection techniques are essential to filter out noise and identify the most informative features, thereby preventing the inclusion of irrelevant information that may degrade model performance.

2.1.2 Advantages of Feature Selection

Among many, the three important advantages of feature selection in building Machine Learning models are mentioned below:

- **Reduces Training Time and Inference Time**

Feature selection provides a significant advantage by streamlining both the training and inference phases of Machine Learning models. By eliminating irrelevant or redundant features, the dimensionality of the dataset is reduced, leading to more efficient computation and faster convergence during the training process. This acceleration is particularly valuable in resource-constrained environments, enhancing the overall efficiency of the model development lifecycle. Feature selection not only accelerates the learning process but also enables faster predictions during real-time inference, contributing to improved computational efficiency throughout the model's lifecycle.

- **Increase Model Interpretability**

Carefully selecting a subset of relevant features enhances the interpretability of Machine Learning models. Feature selection allows practitioners to concentrate on the most influential variables, facilitating a clearer understanding of the underlying relationships between features and the target outcome. This transparency is invaluable in domains where model interpretability is critical, such as healthcare or finance, enabling stakeholders to make informed decisions based on a more comprehensible model structure. By focusing on the most noteworthy features, decision-makers can gain insights into the key drivers of model output, promoting trust and accountability in the decision-making process.

- **Variable Redundancy**

Feature selection plays a critical role by strategically filtering out redundant variables that carry overlapping or highly-correlated information. By retaining only the most informative features, the risk of multicollinearity is mitigated, enhancing the stability and reliability of the model. This process not only simplifies the model's structure but also contributes to a more robust and generalizable predictive framework. By eliminating redundancies, feature selection helps to create a leaner and more efficient model, improving its ability to generalize effectively across diverse data scenarios while maintaining high predictive accuracy.

2.2 Overview of Classical Feature Selection Techniques

Classical feature selection methods are categorized into three types: filter methods, wrapper methods and embedded methods.

- **Filter Methods**

Filter methods are a feature selection technique that selects features based on specific criteria before constructing the model. Filter methods approach feature selection as a pre-processing step, independent of the subsequent learning algorithm. They rely solely on the inherent characteristics of the data and are computationally efficient, inexpensive, and excel at identifying and removing duplicate, correlated, and redundant features.

Set of all features $\longrightarrow$ Selecting the best subset $\longrightarrow$ Learning algorithm $\longrightarrow$ Performance

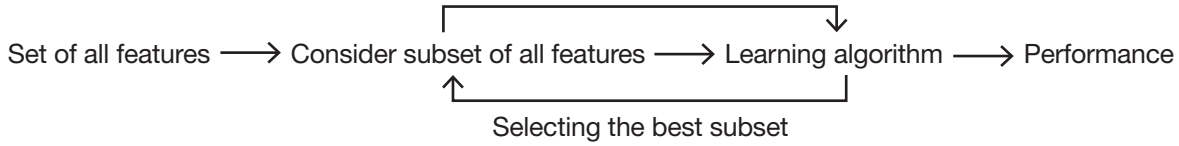
Some of the filter methods are:

- **Information Gain** - Evaluates the worth of a feature by measuring the reduction in entropy, or uncertainty, in the class label distribution when the feature is known.
- **Chi-Square Test** - Assesses the independence between a feature and the target variable, selecting features that have a strong association with the target.
- **Fisher's Score** - Ranks features by calculating the ratio of the variance between different classes to the variance within each class, selecting features that maximize class separability.
- **Correlation Coefficient** - Measures the linear relationship between a feature and the target variable, selecting features that have a high correlation with the target.
- **Variance Threshold** - Removes features with low variance if features with low variance do not contribute significantly to the model's predictive power.

- **Wrapper Methods**

Wrapper methods take a tailored approach to feature selection, aligning the process with the specific Machine Learning algorithm being applied to a given dataset. Utilizing a greedy search strategy, these methods systematically evaluate all combinations of features against a predefined evaluation criterion. This criterion is

determined by the problem and may involve metrics such as p-values, R-squared or Adjusted R-squared for regression problems, or accuracy, precision, recall, or f1-score for classification tasks. Wrapper methods identify the optimal feature combination that yields the best performance for the designated Machine Learning algorithm, ensuring a highly-customized and efficient feature selection process.

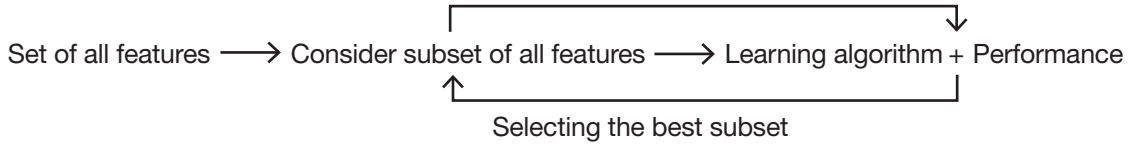


Three commonly used techniques under wrapper methods are:

- **Forward Selection**: Starts with an empty model and iteratively adds features that improve the model's performance the most, continuing until no significant improvement is observed.
- **Backward Elimination**: Begins with a full model containing all features and iteratively removes the least significant features, one at a time, until further removal reduces model performance.
- **Bi-directional Elimination (Stepwise Selection)**: Combines forward selection and backward elimination, adding and removing features in each iteration to find the optimal set that improves model performance.

- **Embedded Methods**

Embedded methods amalgamate the beneficial aspects of both filter and wrapper approaches to feature selection. In embedded methods, the feature selection process is an inherent component of the classification or regression model development. These techniques perform feature selection and algorithm training concurrently, integrating the two processes into a unified framework. By seamlessly intertwining feature selection with model training, embedded methods leverage the strengths of both filter and wrapper methodologies, potentially yielding more efficient and effective outcomes.



Four commonly used techniques under embedded methods are:

- **Least Absolute Shrinkage and Selection Operator (LASSO)** - Performs feature selection by adding an L1 regularization term to the linear regression model, which shrinks some coefficients to zero, effectively removing less important features.
- **Ridge Regression** - Unlike LASSO, Ridge Regression uses an L2 regularization term that penalizes large coefficients but does not perform feature selection directly; instead, it reduces multicollinearity by shrinking coefficients.
- **Tree-based Feature Importance** - Evaluates the importance of features based on how frequently and effectively they are used to split nodes in a decision tree or ensemble methods like Random Forests with important features contributing more to the model's predictive power.
- **Permutation Importance** - Assesses feature importance by randomly shuffling each feature and measuring the increase in the model's error, with a larger increase indicating higher importance of the feature for model predictions.

2.3 Shortcomings of Current Classical Methods

Classical feature selection techniques, such as filter methods, wrapper methods and embedded methods, offer distinct advantages and yet face specific limitations. Filter methods, though computationally efficient, may fail to capture crucial feature interactions essential for certain models, and often disregard dependencies among features. Wrapper methods are computationally intensive due to exhaustive search across the feature space, potentially leading to overfitting and reduced generalization performance. In addition, reliance on a specific Machine Learning algorithm limits the versatility of wrapper methods. Embedded methods, while integrating feature selection into the training process, may be constrained by model-specific limitations and add complexity to the implementation.

A common challenge across the feature selection techniques is sensitivity to hyperparameter settings, which poses a trade-off between computational efficiency and model effectiveness. Suboptimal choices of features

significantly impact overall performance, and therefore feature selection methods require careful tuning of hyperparameters. Therefore, model developers must navigate the trade-offs, considering factors such as available computational resources, dataset characteristics and the specific goals of the Machine Learning task when selecting an appropriate feature selection method.

The choice of feature selection technique requires a careful evaluation of the specific requirements and constraints of the problem at hand, weighing the potential benefits against the associated drawbacks and limitations.

2.4 Overview of Quantum Feature Selection

Quantum-enhanced Machine Learning is one of the primary areas of quantum computing research. The underlying framework of quantum logic maps well to the fields of linear algebra and mathematical optimization which are essential for building algorithms in Machine Learning. In addition to applying quantum computing for designing ML models such as Quantum Neural Networks (QNN), quantum techniques can drive an efficient feature selection process.

In our current work, we frame feature selection as a combinatorial optimization problem, aiming to reduce dimensionality by removing irrelevant, noisy and redundant features while adhering to global constraints. The optimal trade-off between feature relevance and independence can be expressed as a Quadratic Unconstrained Binary Optimization (QUBO) problem with a parameter α $(0 \leq \alpha \leq 1)$ representing the desired balance. This QUBO formulation can then be solved using quantum computing methods, harnessing the unique capabilities of quantum algorithms for enhanced performance and scalability. Additionally, our discussion focuses on the performance and comparative analysis of quantum-based feature selection against existing classical methods, highlighting the potential advantages and implications for real-world applications.

3. Our Solution Approach

As we frame the feature selection problem as a combinatorial optimization problem, a good understanding of solving optimization problems is highly beneficial.

Solving an optimization problem involves two key steps: first, formulating the problem, and second, obtaining the optimal solution to the formulation. The initial step entails understanding the problem and expressing it in a mathematical format. An optimization problem mathematically is formulated in terms of decision variables, objective function and problem constraints.

Decision variables are the variables whose values vary over a set and whose choice determines whether the problem is solved optimally and feasibly. E.g., assignment of a worker to a task. The objective function provides us with the mathematical definition of quantity to be maximized or minimized. In the example of worker assignments, the objective function can be defined in terms of worker productivity. Problem constraints define the rules that should be satisfied when solving the optimization problem. Taking the working example again, one constraint could be the number of hours per day a worker is employed.

This mathematical formulation can be achieved through various methods, such as Linear Programming (LP), Mixed Integer Linear Programming (MILP), non-linear or quadratic approaches. The choice of formulation method depends on the nature of the problem, the convenience of expressing it mathematically, and the availability of algorithms, techniques and tools to solve the resulting model.

Once the problem is formulated, the second step involves obtaining the solution with the most optimal cost. As problem size increases, analytical solutions become impractical, and brute-force methods can become exponentially more time-consuming. Consequently, numerical approaches are employed to provide approximate solutions to large-scale optimization problems efficiently.

We delve into these two steps for our feature selection problem in section 3.1 and 3.2.

3.1 Optimization Problem Formulation

The primary objective of feature selection methods is to identify the most relevant features, either individually or in combination while discarding redundant and irrelevant ones. This process aims to preserve the essential information contained within the complete set of input variables concerning the target class. To measure relevancy and redundancy effectively, evaluation measures play a crucial role. We leverage information theory measures to quantify relevance, redundancy and the cooperativeness of features.

Several sets of information measures can be employed to evaluate relationships within features and with the target variable. Selecting the appropriate measure depends on the presence of specific feature types, such as categorical and numerical features, in the dataset and the consistency of the information measures in evaluating feature relationships.

We have identified Mutual Information (MI) as the most suitable information measure for general Machine Learning datasets. Mutual information, also known as Information Gain (IG) or two-way interaction, quantifies the stochastic dependency between variables, making it an effective bivariate measure of correlation. By leveraging mutual information as the evaluation measure, we can effectively identify the most relevant features while discarding redundant and irrelevant ones, enhancing the performance and interpretability of Machine Learning models.

MI between continuous variables X and Y can be defined by I,

$$I(Y,X) = \int_{\Omega_Y} \int_{\Omega_X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$$

where Ω_Y and Ω_X are the sample spaces corresponding to Y and X, p (x, y) is the joint probability density, and p(x), p(y) are the marginal density function.

For discrete variables Y and X, the MI formula takes the form:

$$I(Y,X) = \sum_{y \in \Omega_Y} \sum_{x \in \Omega_X} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

While estimating Mutual Information (MI) between features, various heuristics are employed based on data types and sample sizes to obtain robust estimates. For continuous and discrete variable sets, nearest-neighbor and distance-based measures can be utilized. Alternatively, binning or discretization of continuous data can be applied to calculate MI using established formulas.

Armed with the above knowledge, we define the decision variables, objective function and problem constraints for our problem below.

Decision Variables

In our case, decision variables are binary variables indicating the selection of a feature in the optimal list. Given a dataset of N+1 dimension where 'N' is the number of independent and an additional target variable, subsequently leads to 'N' binary decision variables.

Objective Function

Our optimization objective is to maximize feature relevance to the target variable while minimizing redundancy within features. We combine the obtained relevance and redundancy information, defining the overall objective as a maximization problem. The trade-off between relevance and redundancy is controlled by the hyperparameter **'α'**. To account for redundancy overflow, computed by considering **(N*N – N)**) interactions between independent features (excluding N self-associations), compared to relevance computed from **N** interactions with the target, we divide overall redundancy by S, the total number of optimal features. This formulation aims to achieve maximum relevance and minimum redundancy, striking the desired balance through the **'α'** parameter's adjustment.

Problem Constraints

In our case, the total number of features to be selected should not be greater than the desired number 'S' is considered as a constraint.

Finally, we mathematically define two optimization formulations:

- Maximum Relevancy and Minimum Redundancy

$$\max_S \left[\alpha \sum_{i \in S} I(X_i;Y) - \left(\frac{1-\alpha}{S}\right) \sum_{i,j \in S, j \neq i} I(X_i;X_j) \right]$$

In addition, we propose another type of objective formulation to check for maximum relevancy and maximum complement of information among the features, mentioned below:

- Maximum Complement and Maximum Relevancy

$$\underset{S}{\operatorname{argmax}} \sum_{i \in S} \left[I(X_i; Y) + \sum_{j \in S, j \neq i} I(X_j; Y \mid X_i) \right]$$

3.2 Solving the Optimization Problem

Solving an optimization problem with the decision variables, constraints and a quadratic objective function requires a non-linear solver. Here we make use of hybrid quantum annealers from D-wave while also comparing it against a classical algorithm which can be used to solve such formulations. Traditional classical approaches other than simulated annealing such as feature importance and permutation importance are explored in our datasets. Below we provide a brief overview of quantum and simulated annealing.

Classical Optimization Approach

The problem is formulated as a Binary Quadratic Model (BQM) using the dimod package from D-Wave's Ocean SDK. Simulated Annealing is applied to solve the problem. D-Wave's Simulated Annealing Sampler (SAS) is a classical algorithm commonly used in heuristic optimization problems and approximate Boltzmann sampling, well suited to finding solutions for large problems.

Quantum Annealer-based Approach

Quantum annealers are ISING machines that help solve combinatorial optimization problems. Solving optimization problems with quantum annealers requires encoding problems to energy minimization problems. Quantum annealers employ energy encoding to map problems to hardware and follow a nature-inspired quantum optimization paradigm. It allows the system to evolve through time while maintaining control over the pace of evolution, and when given enough time, a system will achieve its lowest energy point.

While classical algorithms, such as simulated annealing, also employ a similar phenomenon, quantum annealers can deliver significant performance and quality improvement over classical algorithms using quantum mechanical phenomena such as quantum tunneling.

D-Wave's quantum annealers currently support optimization models in the form of Constrained Quadratic Models (CQM) or Binary Quadratic Models (BQM) to define objectives and constraints. BQMs are further transformed into Quadratic Unconstrained Binary Optimization (QUBO) or equivalent ISING formulation in ferromagnetism. To use quantum annealers, first, the optimization problem must be converted to CQMs or QUBOs (Quadratic Unconstrained Binary Optimization). CQM, as the name suggests, are constrained models with binary or integer decision variables. QUBOs have only binary decision variables, and the constraints need to be converted into an unconstrained problem using the penalty method. In this method, constraints are added to the objective function with a penalty. If a solution fails to satisfy a constraint, then the corresponding penalty will be added to the total cost.

Dimod has been utilized for CQM formulation, while Qubovert has been utilized to formulate the QUBO model. D-Wave's Leap Hybrid Solvers, which are used to solve the problem, implement state-of-the-art classical algorithms together with intelligent allocation of the quantum computer to parts of the problem where it benefits most.

4. Experiments & Results

For experimentation and testing of our approach, we utilize two high-dimensional datasets. The performance of quantum feature selection is assessed against three sets of feature selection methods, namely simulated annealing, and classical feature selection methods such as feature importance and permutation importance. We also evaluated the performance of QFS against the wrapper method - Scikit-learn's Sequential Feature Selector. However, the time required for feature selection was substantial, ~ 45-60 minutes for an optimal feature subset of 5 to 20 features. Therefore, the results and discussion exclude wrapper methods.

We utilize the above-mentioned feature selection methods, train a random forest ML model and log the set of classification evaluation metrics such as F1-score, accuracy, AUC-ROC score, etc. The subsequent sections describe the use case, their dataset description and the results obtained.

Malicious Software Detection Dataset

In the contemporary digital landscape, the proliferation of malicious software, commonly known as malware, poses a severe threat to the integrity, confidentiality and availability of information systems. The ability to detect malware in real-time is imperative for preventing potential damage to systems and data.

The "TUNADROMD" dataset focuses on the intricate task of distinguishing between malicious software (malware) and legitimate software (goodware). The dataset consists of 4464 instances and 241 features and represents information on 80% of malware class labels and the remaining 20% about goodware. These attributes include various characteristics of software files or behaviors, such as file size, code patterns, permissions and many other factors that are relevant for distinguishing between malware and legitimate software. We utilize this dataset for malware software detection tasks and evaluate the performance of the random forest Machine Learning model utilizing different feature selection methods.

TUNADROMD Dataset	
Number of Variables	242
Number of Observations	4465
Variable Type - Numeric	2
Variable Type - Categorical	240

Parkinson's Disease Detection Dataset

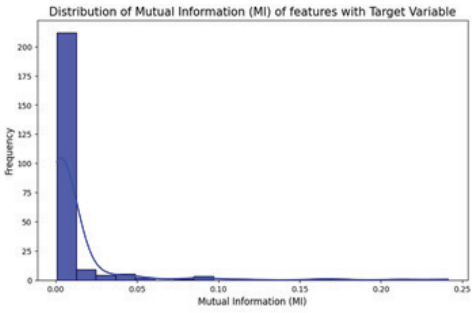
Parkinson's Disease (PD) is one of the most common neurodegenerative diseases with a high prevalence rate. The detection of PD-positive subjects is vital in terms of disease prognosis, diagnostics, management and treatment. Different types of early symptoms, such as speech impairment and changes in writing, are associated with Parkinson's disease.

The PD speech signal dataset consists of speech features extracted from 188 patients and sixty-four healthy controls, using a variety of speech signal processing techniques. It comprises of 756 instances and 754 features for the 75% class label of patients with Parkinson's disease and the remaining 25% with no disease. We utilize the dataset for Parkinson's disease classification and evaluate the performance of the Machine Learning model, as described in earlier sections.

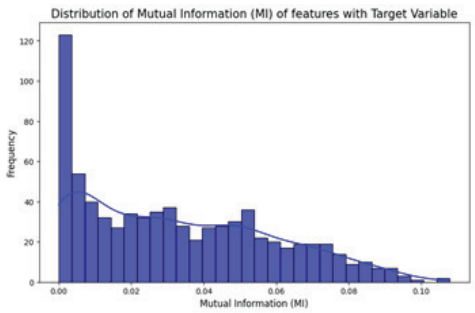
Parkinson's Disease Dataset	
Number of Variables	755
Number of Observations	756
Variable Type - Numeric	753
Variable Type - Categorical	2

4.1 Performance Results

- Distribution of Mutual Information of Features with the Target Variable

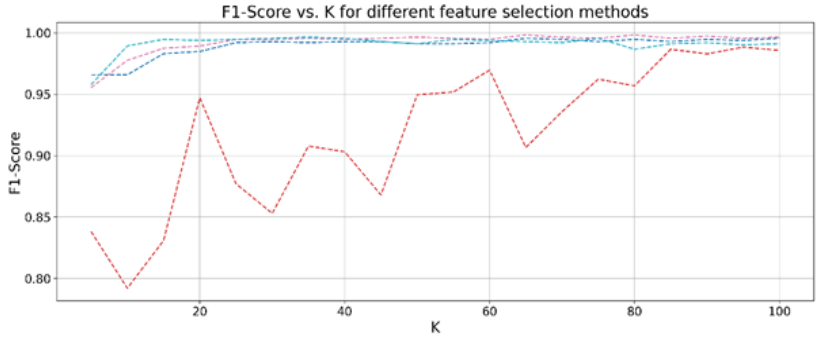


TUANDROMD Dataset

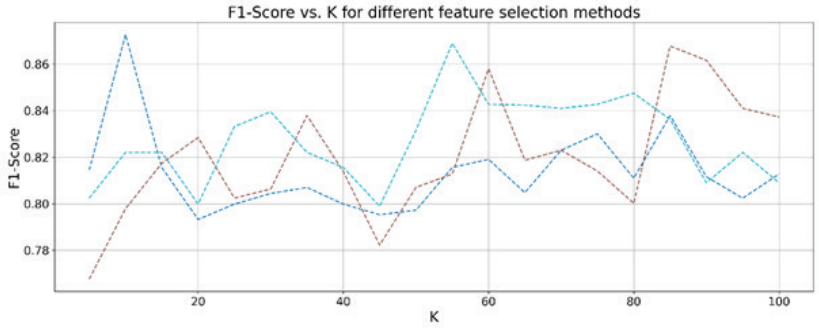


Parkinson's Disease (PD) Dataset

- F1-Score vs. Feature subset cardinality for different Feature selection methods

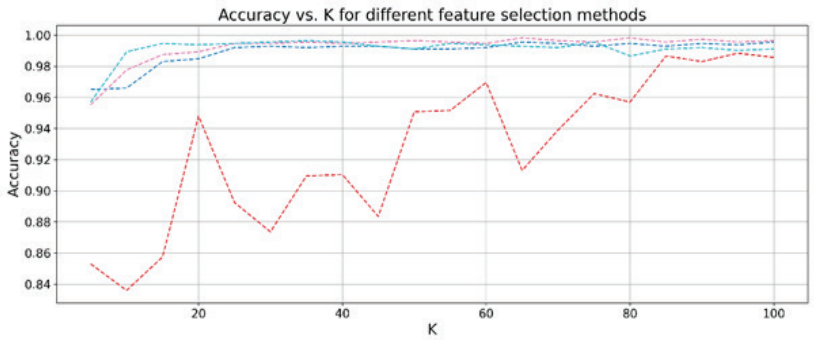


TUANDROMD Dataset

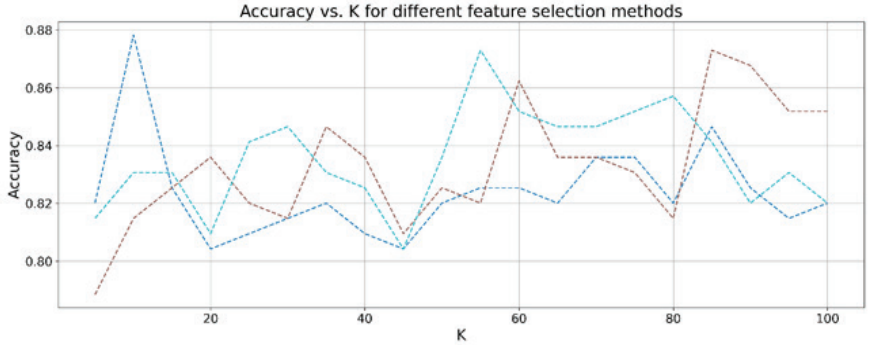


Parkinson's Disease Dataset

- Accuracy vs. Feature subset cardinality for different feature selection methods

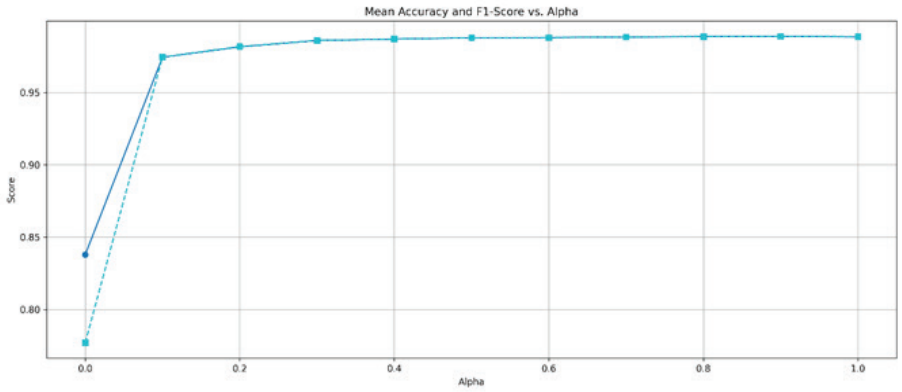


TUANDROMD Dataset

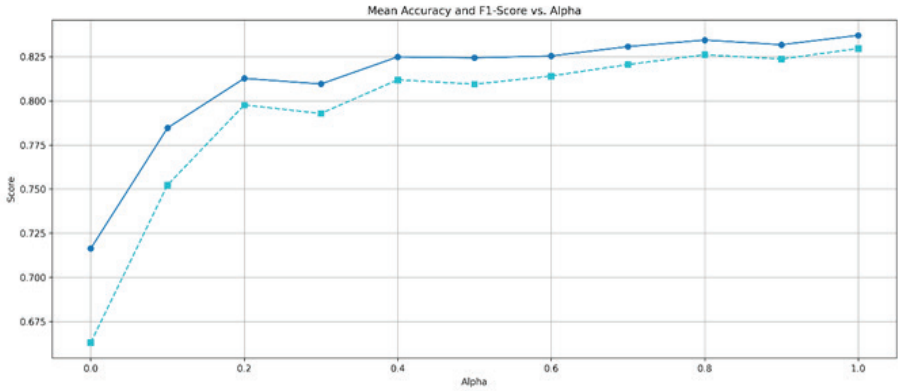


Parkinson's Disease Dataset

- Accuracy & F1-Score vs. Alpha for QFS with varying Feature subset cardinality (K)

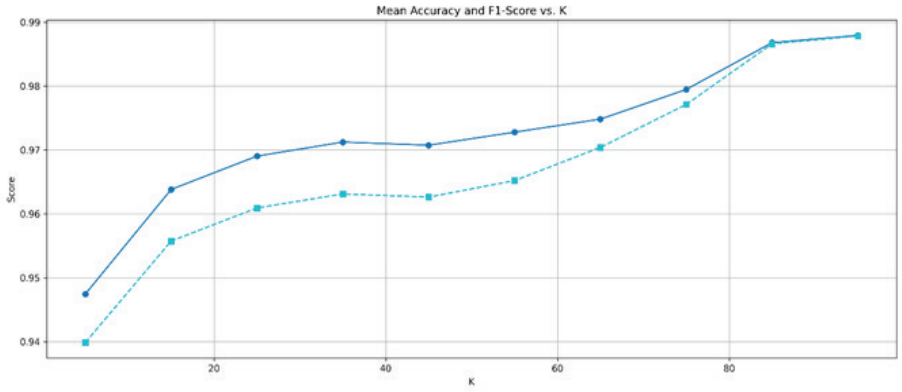


TUANDROMD Dataset

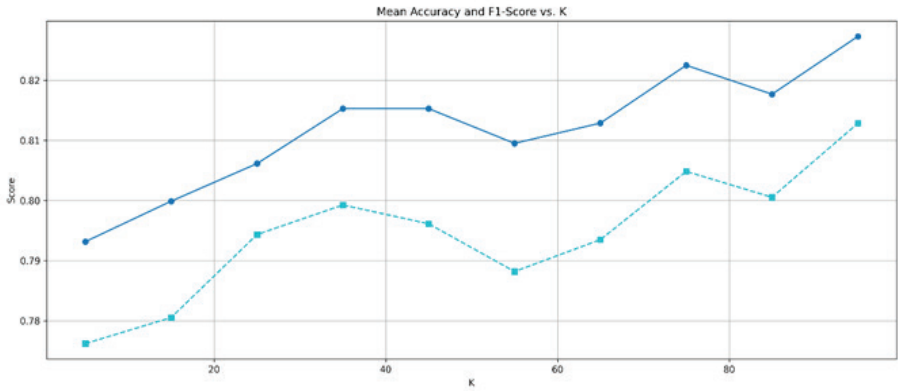


Parkinson's Disease Dataset

- Accuracy & F1-Score vs. Feature subset cardinality (K) for QFS with varying Alpha



TUANDROMD Dataset



Parkinson's Disease Dataset

5. Analysis of Results

- **Quality of Solution**
  - For accuracy and F1-score on the TUANDROMD dataset, the region between 20 to 100 feature subset the results do not differ by the spread of accuracy across feature selection methods, suggesting comparable performance across all methods.
  - As per the performance charts in Section 4.4, the performance score for the lower range of cardinality of the feature subset is better for QFS compared to other feature selection methods for both datasets. This suggests a good learning ability of QFS with limited data and feature sets.
  - In the results, we evaluate the mean accuracy and mean F1-score of the random forest model across a range of cardinality of feature subset and alpha values (trade-off between relevancy and redundancy). For the dataset used in our experiments, we witness an increase in performance KPIs with an increase in alpha values. However, this trend may depend on the nature of the dataset.
  - As per the above analysis, we also assess the mean accuracy and mean F1-score for different feature subsets and witness an increase in performance KPIs with an increase in the cardinality of the optimal feature subset.
  - We also observe a crucial role of two hyperparameters in our algorithm: **Feature subset cardinality (K)** and **Relevancy to Redundancy ratio (Alpha).** For the TUANDROMD dataset, optimal performance is achieved at an Alpha value between 0.7 and 0.9. The relevancy of features plays a critical role in attaining optimal performance for Machine Learning algorithms, highlighting the importance of this hyperparameter. This finding encourages us to explore alternative formulations that prioritize maximum relevancy and maximum complementarity among features, aiming to further enhance the algorithm's performance. By adjusting and carefully tuning these hyperparameters, we aim to achieve a more refined and effective feature selection process.

- **Scalability of Solution**
  - The CQM quantum annealer provides a scalable solution. For the optimization problem we formulated, quantum annealers demonstrate superior performance compared to the simulated annealing solver for both problem types, which involve 241 and 754 decision variables, respectively.

6. Conclusion

In this whitepaper, we have explored the potential of Quantum Feature Selection (QFS) as a powerful tool for addressing the challenges in high-dimensional datasets in Machine Learning. By framing feature selection as a combinatorial optimization problem and leveraging quantum computing methods, we demonstrated how QFS can effectively reduce dimensionality while balancing feature relevance and independence. Our approach not only aligns with the core principles of quantum computing but also offers a scalable and efficient alternative to classical feature selection methods.

Through experimentation and analysis, we evaluated the performance of QFS against traditional methods, including simulated annealing and classical feature selection techniques such as feature importance and permutation importance. The results reveal that QFS exhibits a strong learning ability, particularly with limited data and feature sets, and outperforms other methods in the lower range of feature subset cardinality. Furthermore, the hyperparameters of feature subset cardinality and the relevancy-to-redundancy ratio (Alpha) were found to play crucial roles in optimizing the performance of QFS.

The scalability of our solution was further validated by the performance of the CQM quantum annealer, which demonstrated superior results compared to simulated annealing across different problem types involving large decision variables. These findings highlight the significant potential of quantum computing in enhancing feature selection processes, offering a promising avenue for future research and application in real-world scenarios.

Overall, our work underscores the importance of quantum-enhanced techniques in advancing Machine Learning methodologies, paving the way for more efficient, scalable and effective solutions to complex problems in the field.

7. Next Steps

The next steps in our exploration of quantum computing for high-dimensional Machine Learning involve several key avenues for further research and evaluation. First, combine QFS with Quantum Machine Learning (QML) techniques to assess and benchmark the performance improvements brought by this integration. Second, investigate robust and consistent measures for estimating mutual information between variables, potentially leveraging quantum methods for this estimation. Our preliminary tests suggest that managing the trade-off between relevancy and redundancy is crucial for optimizing performance. Third, evaluate the performance of the conditional mutual information-based optimization formulation proposed in this paper, which has not yet been evaluated. This comprehensive approach will provide a deeper understanding of the capabilities and limitations of QFS in various scenarios.

8. References

- Optimal feature selection in credit scoring and classification using a quantum annealer - Andrew Milne, Maxwell Rounds, and Phil Goddard: 1Qbit
- Evaluation of Feature Selection Methods: Farhan Mohammad and Samira Golsefid, Ph.D.: Paypal
- Optimal Feature Discovery: Better, Leaner Machine Learning Models Through Information Theory: Uber
- Exploratory Data Analysis for Feature Selection in Machine Learning: Google WhitePaper
- Exploratory data analysis, feature selection for better ML models - Luo Shixin, Cloud Machine Learning Engineer: Google blog
- Dr.S.Sumathi. (2023). TUNADROMD- Malware Detection. Kaggle. https://kaggle.com/competitions/tunadromd-malware-detection
- Sakar, C., Serbes, G., Gunduz, A., Nizam, H., & Sakar, B. (2018). Parkinson's Disease Classification [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5MS4X.
- D-Wave Systems. (n.d.). D-Wave scikit-learn Plugin. GitHub. https://github.com/dwavesystems/dwave-scikit-learn-plugin

About Mphasis

Mphasis' purpose is to be the *"Driver in the Driverless Car"* for Global Enterprises by applying next-generation design, architecture and engineering services, to deliver scalable and sustainable software and technology solutions. Customer centricity is foundational to Mphasis, and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized ($C = X2C^{2}_{TM} = 1$) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization, combined with an integrated sustainability and purpose-led approach across its operations and solutions are key to building strong relationships with marquee clients. Click here to know more. (BSE: 526299; NSE: MPHASIS)

For more information, contact: marketinginfo.m@mphasis.com

USA
Mphasis Corporation
41 Madison Avenue
35th Floor, New York
New York 10010, USA
Tel: +1 (212) 686 6655

UK
Mphasis UK Limited
1 Ropemaker Street, London
EC2Y 9HT, United Kingdom
T : +44 020 7153 1327

INDIA
Mphasis Limited
Bagmane World Technology Center
Marathahalli Ring Road
Doddanakundhi Village, Mahadevapura
Bangalore 560 048, India
Tel.: +91 80 3352 5000

www.mphasis.com

VAS 12/12/24 US LETTER BASIL 9280