JURNAL TEKNIK INFORMATIKA DAN MULTIMEDIA Vol 3, No.1, Mei 2023, pp. 38 - 58
p-ISSN : 2808-909X (print)
e-ISSN : 2798-2505 (online)
http://journal.politeknik-pratama.ac.id/index.php/JTIM · page 38

EKSPLORASI KAPASITAS PENGKODEAN AMPLITUDO UNTUK MODEL QUANTUM MACHINE LEARNING

Rabiatul Adawiyah[1], Munifah[2]

[1] Program Studi Komputerisasi Akuntansi Universitas Sains dan Teknologi Komputer
Jl. Majapahit 605 Semarang, telp : (024)-6723456, e-mail: rabiatul@stekom.ac.id
[2] Program Studi Komputerisasi Akuntansi Universitas Sains dan Teknologi Komputer
Jl. Majapahit 605 Semarang, telp : (024)-6723456, e-mail: munifah@stekom.ac.id

ARTICLE INFO

Article history:

Received 10 April 2023
Received in revised form 24 April 2023
Accepted 8 Mei 2023
Available online 15 Mei 2023

ABSTRACT

Computing performs calculations using physical phenomena and mechanical principles needed to solve problems. This form of processing has theoretically been shown to provide speed to some modern processing problems. New developments in technological innovation are emerging, and the application of learning models for these new devices is growing. With much anticipation the hunting phenomenon in the field of Machine Learning has become clear. This research develops a TensorFlow Quantum (TFQ) software framework model for machine learning purposes. The two models developed utilize information coding techniques from amplitude coding to construct situations in the learning model. The aim of this study was to explore the capacity of amplitude coding to prepare preparatory states enriched in learning methods and in-depth analysis of data properties that provide insight into the training data using the Variational Quantum Classifier (VQC). The emergence of this new method raises the question of how best to use this tool, the aim is to provide some overview explanations for the applied machine learning state that can be applied given the constraints of the actual device. The results of this study indicate there are clear advantages to using amplitude coding over other methods such as using hybrid classical destructive neural networks in TFQ. In addition, there are several pre-processing steps that can generate more feature-rich data when using VQC, basically the free lunch theorem is not as valid for the learning to sell method as it is for the classical technique. Information even though encoded in modified form does not change the data preparation steps but involves new ways of understanding and appreciating these new methods. Future work in this area will require expansion into multiclass classification techniques that are sufficiently developed to be applicable in work such as this.

Keywords: Machine Learning, Pengkodean Amplitudo, Quantum Machine Learning, Quantum Computation, TensorFlow Quantum.

Received April 10, 2023; Accepted Mei 8, 2022; Available online Mei 15, 2023

p-ISSN : 2808-909X e-ISSN : 2798-2505

Abstrak

Komputasi kuantum melakukan perhitungan dengan menggunakan fenomena fisik dan prinsip mekanika kuantum untuk memecahkan masalah. Bentuk komputasi ini secara teoritis telah terbukti memberikan percepatan pada beberapa masalah pemrosesan modern. Perkembangan baru dalam teknologi kuantum mulai bermunculan, dan penerapan model pembelajaran untuk perangkat baru ini terus berkembang. Dengan banyak antisipasi pemanfaatan fenomena kuantum di bidang Machine Learning telah menjadi jelas. Penelitian ini mengembangkan model kerangka kerja perangkat lunak TensorFlow Quantum (TFQ) untuk tujuan machine learning. Kedua model yang dikembangkan memanfaatkan teknik pengkodean informasi dari pengkodean amplitudo untuk penyusunan keadaan dalam model pembelajaran kuantum. Tujuan dari penelitian ini adalah mengeksplorasi kapasitas pengkodean amplitudo untuk menyediakan persiapan keadaan yang diperkaya dalam metode pembelajaran dan analisis mendalam tentang properti data yang memberikan wawasan tentang data pelatihan menggunakan Variational Quantum Classifier (VQC). Munculnya metode baru ini menimbulkan pertanyaan tentang cara terbaik menggunakan alat ini, tujuannya adalah untuk memberikan beberapa penjelasan ikhtisar untuk keadaan machine learning kuantum yang dapat diterapkan mengingat kendala perangkat yang sebenarnya. Hasil penelitian ini menunjukkan ada keuntungan yang jelas untuk menggunakan pengkodean amplitudo dibandingkan metode lain seperti ditunjukkan menggunakan jaringan saraf klasik kuantum hibrid di TFQ. Selain itu, ada beberapa langkah prapemrosesan yang dapat menghasilkan lebih banyak data kaya fitur saat menggunakan VQC, pada dasarnya teorema makan siang tidak gratis berlaku untuk metode pembelajaran kuantum seperti halnya dalam teknik klasik. Informasi meskipun dikodekan dalam bentuk kuantum tidak mengubah langkah-langkah penyiapan data tetapi melibatkan cara-cara baru untuk memahami dan mengapresiasi metode-metode baru ini. Pekerjaan masa depan di bidang ini akan membutuhkan perluasan ke dalam teknik klasifikasi multikelas yang cukup dikembangkan untuk dapat diterapkan dalam pekerjaan seperti ini..

Kata Kunci: Machine Learning, Pengkodean Amplitudo, Quantum Machine Learning, Quantum Computation, TensorFlow Quantum.

1. PENDAHULUAN

Quantum Machine Learning (QML) adalah bidang interdisipliner yang menggabungkan antara Quantum Computing (QC) dan Machine Learning (ML). Saat ini, ada banyak algoritma yang muncul di bidang QC algoritme dan mulai terlihat jelas karena perangkat yang diciptakan Noisy Intermediate Scale Quantum (NISQ) telah menjadi implementasi asli dari Quantum Processing Units (QPU) (Travis et al., 2017). Perangkat NISQ membuka pintu untuk proses pengembangan algoritma, ini juga meningkatkan minat seputar topik, secara signifikan menjangkau domain di luar penelitian komputasi biasa seperti keuangan, kimia, dan biologi (Havel et al., 1997; Bob et al., 2016; Lin et al., 2017). Perangkat era NISQ memberikan kesempatan unik untuk menguji dan mengembangkan teknik QML yang secara fisik tidak mungkin dilakukan sebelumnya. Perangkat era NISQ telah menghasilkan pengembangan beberapa suite pemrograman untuk perangkat kuantum yang disimulasikan.

Bersama dengan kekuatan pemrosesan modern, perangkat era NISQ memungkinkan pengalaman yang jauh lebih kaya dalam eksperimen perangkat. Dalam hal Application Programable Interfaces (APIs) penelitian ini memiliki dua komponen utama yaitu kode pendukung untuk pengembangan sirkuit dan kode kompilasi yang berjalan di perangkat fisik. Organisasi besar seperti IBM, dan Google memungkinkan pengembangan model pada perangkat kuantum simulasi dan nyata. Rangkaian perangkat lunak ini memungkinkan penerapan gerbang agnostik perangkat mendasar yang dapat digunakan untuk membangun algoritme kuantum. Dengan kode pengembangan maka sirkuit dan oracle dapat dibuat diluar gerbang. Oracle dianggap sebagai "kotak hitam" dalam sirkuit kuantum yang menerapkan beberapa set gerbang untuk melakukan perubahan pada dasar komputasi keadaan kuantum (Brassard ey al., (1994); Zhandry et al., (2011)). Beberapa dari bahasa ini (Qiskit Aer dan Cirq) dibuat dan didukung oleh masing-masing IBM, Google, dan Xanadu. Ada juga yang bisa dikatakan "backend" yang perlu menangani masalah perangkat lunak dan fisik. Backend menangani aspek-aspek seperti, transpiling kode, pemetaan qubit ke topologi perangkat, dan derau selama pelaksanaan percobaan. Masing-masing kumpulan kode ini menangani langkah-langkah backend ini dengan caranya sendiri.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Ada beberapa perusahaan lain yang juga bergabung dan unggul dalam perlombaan di ruang QC. Dalam beberapa kasus, perusahaan-perusahaan ini mengembangkan QC mereka sendiri sementara yang lain mengembangkan rangkaian perangkat lunak seperti Amazon dengan BraKet, Microsoft dengan Q# dan Honeywell yang menggunakan Quantum Assembler (QASM) API. Industri mulai tunduk pada perlombaan kuantum ini karena proyeksi menunjukkan supremasi kuantum tepat di depan mata. Beberapa pemain utama di QML telah mengembangkan kerangka kerja sumber terbuka seperti Qiskit & Q Experience IBM, dan TensorFlow Quantum Google. Ketiganya digunakan dalam pekerjaan yang disajikan dalam penelitian ini karena mereka memiliki paket, dukungan, dan komunitas yang paling luas yang terlibat untuk QML. Masing-masing kerangka kerja ini memiliki kemampuan untuk memperluas rangkaian perangkat lunak mereka dengan memanfaatkan APIs tersier serta memanfaatkan kerangka kerja mereka pada perangkat di luar perusahaan masing-masing. Sebagian besar pekerjaan yang disajikan dalam penelitian ini berkaitan dengan pemrosesan sebelum dan sesudah metodologi pembelajaran diterapkan, salah satu persyaratan paling mendasar untuk memanfaatkan QML, atau ML dalam hal ini, adalah fondasi prosedur penambangan data umum. Beberapa metode yang akan dikembangkan bersifat klasik. Satu metode kuantum yang digunakan di sini dikenal sebagai pengkodean amplitudo. Metode ini seperti jauh lebih unggul daripada metode pengkodean data klasik mentah lainnya.

Data mentah dari suatu sistem atau perangkat lunak sering diambil tanpa dalih apa pun selain tujuan penggunaannya yang langsung pada sejumlah bentuk berbeda yang dapat diambil data seperti variabel nyata kontinu, variabel diskrit kategoris, variabel biner, teks tidak terstruktur, multimedia (audio, gambar, dan video). Dalam bentuk mentahnya, data seringkali belum siap untuk dipelajari atau digunakan secara analitis, bahkan maknanya sangat sulit dipahami tanpa beberapa tingkat klarifikasi (Mehmed et al., (2011)). Tiga masalah yang sering dikaitkan dengan pemrosesan data melibatkan catatan yang hilang, kelas yang tidak seimbang, dan outlier. Catatan yang hilang menimbulkan masalah yang sangat sulit karena catatan tersebut berisi informasi yang akan berharga untuk teknik pembelajaran. Ada beberapa metode untuk menangani data yang hilang seperti menggantinya dengan rata-rata fitur, tetapi ini juga menyebabkan hilangnya konten yang berpotensi berguna. Kelas yang tidak seimbang juga menjadi masalah utama saat menangani data terutama pada pembelajaran heuristik. Ketidakseimbangan kelas terjadi ketika satu set kelas atau satu kelas memiliki lebih banyak sampel daripada kelas lainnya. Sederhananya populasi satu kelas lebih besar dari yang lain.

Dalam kasus ekstrim ada kemungkinan teknik hanya akan menilai semua data sebagai kelas mayoritas atas minoritas (Mehmed et al., (2011)). Untuk mengatasi masalah ini, data dapat diseimbangkan hanya dengan menjatuhkan catatan dari kelas mayoritas agar sesuai dengan kelas minoritas. Selain itu, metode untuk oversampling kelas minoritas telah dikembangkan seperti Synthetic Minority Oversampling Technique (SMOTE) dan Tomek Links (Philip et al., (2002); Matwinet al., (1997)). Isu outlier ketiga seringkali lebih kompleks dalam analisis dan penanganannya. Outlier ada ketika ada data di tepi distribusi sampel. Sederhananya data yang tidak sesuai dengan perilaku umum data yang sama (Mehmed et al., (2011)). Dalam kasus yang paling sederhana, hal ini dapat ditangani dengan mengecualikan sampel yang lebih besar dari sejumlah standar deviasi. Dalam penelitian ini ketiga masalah dalam situasi yang berbeda ditangani dengan baik. Dataset Scikit-Learn yang dihasilkan tidak memiliki banyak masalah, tetapi yang lain seperti dataset Wine menghadapi beberapa masalah ini. Diskritisasi fitur adalah metode untuk mengubah variabel kontinu mentah menjadi ruang fitur diskrit dan kurang kompleks. Secara umum, ini dapat dianggap sebagai pengurangan nilai. Metode paling sederhana untuk diskritisasi adalah mengurutkan data dan pembagiannya berdasarkan bin berukuran m di mana m adalah jumlah elemen dalam bin. Nilai bin kemudian menjadi rata-rata dari setiap bin dan kemudian nilainya dikonversi ke nilai bin tersebut. Metode ini kesulitan menemukan ukuran m yang dibutuhkan untuk mencapai hasil terbaik dan memerlukan beberapa iterasi trial and error (Mehmed et al., (2011)).

Normalization Method	**Standard Deviation**	**Decimal**	**Min-Max**
Equation	$v_i - \text{mean}(v)$	$\frac{v_i}{10^k}$	$\frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)}$

Tabel 1. Metode normalisasi untuk penskalaan data.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Normalisasi adalah salah satu transformasi yang paling umum, namun penting diterapkan pada data. Data skala normalisasi antara beberapa rentang yang telah ditentukan seperti [0, 1]. Nilai disusun berdasarkan beberapa metode yang mempertimbangkan semua sampel untuk satu fitur. Oleh karena itu, normalisasi terjadi secara kolom di seluruh dataset jika berlaku (Mehmed et al., (2011)). Konsep normalisasi adalah untuk menyederhanakan dampak nilai singular apa pun terhadap bias perilaku teknik pembelajaran saat mengamati sampel. Jika suatu teknik melihat bahwa dalam kasus-kasus tertentu suatu nilai sangat besar/kecil, teknik tersebut mungkin terlalu berlebihan/kurang menimbang pentingnya nilai fitur dan menyesatkan proses pembelajaran. Sebelum normalisasi, outlier data harus dihilangkan karena dapat menyebabkan normalisasi mengurangi sebagian besar data menjadi interval nilai yang kecil. Meskipun normalisasi mengurangi interval nilai ke beberapa rentang, jika outlier disertakan, teknik pembelajaran mungkin akan lebih sulit memahami fitur lain dari data. Jika mereka disertakan, mereka juga mempertimbangkan skala untuk normalisasi. Tanpa menghilangkan outlier menggunakan normalisasi dapat menyebabkan kesalahan yang menjadi lebih sulit untuk dipahami lebih lanjut dalam alur analisis. Normalisasi memiliki beberapa bentuk seperti normalisasi standar deviasi, normalisasi desimal, dan normalisasi minimum-maksimum (min-max). Persamaan ini ditunjukkan pada Tabel 1. Dimana kolom yang dinormalisasi adalah v dan nilai kolom adalah vi. Nilai k dalam normalisasi desimal adalah daya yang dibutuhkan untuk membuat nilai terbesar dalam kolom kurang dari atau sama dengan satu dan std adalah standar deviasi. Masing-masing metode ini dapat dimodifikasi agar sesuai dengan data dengan tepat.

Analisis Pasca

Di hampir setiap kasus, cara terbaik untuk melakukan analisis metode pembelajaran dalam ranah klasik maupun kuantum adalah pada kumpulan sampel uji yang belum pernah digunakan di mana pun dalam proses pembelajaran. Ini mengharuskan dataset dibagi menjadi setidaknya dua kelompok seperti pelatihan dan pengujian (Mehmed et al., (2011)). Set validasi akan digunakan untuk memvalidasi dan memberikan umpan balik ke metode pembelajaran saat sedang aktif belajar. Pengujian pada set validasi tidak boleh dilakukan karena teknik akan mengetahui rahasia set ini. Holdout atau set pengujian lagi tidak boleh digunakan oleh model. Holdout set juga harus disiapkan menggunakan metode yang sama dengan data pelatihan. Ini berarti langkah preprocessing yang sama seperti normalisasi dan diskritisasi juga harus diterapkan. Dari sudut pandang implementasi praktis, yang terbaik adalah memisahkan set pengujian dari data segera sebelum memulai pelatihan apa pun, dengan cara ini dapat dipastikan bahwa semua langkah yang diperlukan telah diterapkan dengan benar.

Metrik yang paling umum digunakan di sini adalah Keakuratan model pembelajaran pada set pengujian baru. Dalam penelitian ini, Precision, Recall, F1-Score, Confusion matrixs (True Positive, True Negative, False Positive, dan False Negative), dan Receiver Operator Curves (ROC) juga dipertimbangkan saat mengevaluasi hasil model. Alasan utama menggunakan akurasi karena penelitian ini bekerja dengan data sintetik (Mehmed et al., (2011)). Karena kurangnya kelainan pada data seperti ketidakseimbangan dan outlier, sehingga keakuratannya tidak terlalu dipertanyakan, bukan berarti semua data yang diproses menggunakan metode yang disebutkan di atas tidak dievaluasi, karena perilaku dataset yang diguanakan dalam penelitian ini. Dalam banyak kasus, hasil masih dievaluasi menggunakan semua metrik. Analisis pasca data dapat menjadi rumit ketika hasil dari teknik pembelajaran dijelaskan. Dalam penelitian ini tujuan umum dari analisis pasca adalah untuk menunjukkan bentuk pengkodean dan transformasi mana yang menghasilkan kinerja yang lebih baik dalam hal kemampuan model untuk mempelajari data dalam ruang/representasi kuantum. Untuk alasan ini, alat analisis pasca di kembangkan dan dijelaskan sepenuhnya di bagian Eksperimen.

Machine learning

Metode pembelajaran di zaman modern telah berkembang sangat kompleks dengan perangkat keras baru yang membuka jalan bagi Deep learning dan kemajuan dalam Kecerdasan Buatan. Mengingat kemajuan ini, banyak metode dasar untuk belajar tetap sama. Dalam arti luas, teknik machine leaning dapat dikategorikan menjadi tiga jenis: prediksi, klasifikasi, dan pengelompokan. Secara umum, prediksi adalah tugas yang bertujuan untuk menentukan dengan beberapa tingkat ketepatan atau akurasi suatu nilai yang diberikan sekumpulan input. Baik prediksi maupun klasifikasi berkaitan dengan akurasi dengan cara yang serupa, tetapi akurasi prediksi diukur terhadap hasil prediksi langsung. Klasifikasi bertujuan untuk menentukan kelas sampel mana yang sesuai ketika diidentifikasi. Akurasi dalam hal model klasifikasi ditentukan berdasarkan kumpulan sampel yang diklasifikasikan dengan benar. Metode pengelompokan biasanya didasarkan pada beberapa jenis metrik jarak yang

p-ISSN : 2808-909X e-ISSN : 2798-2505

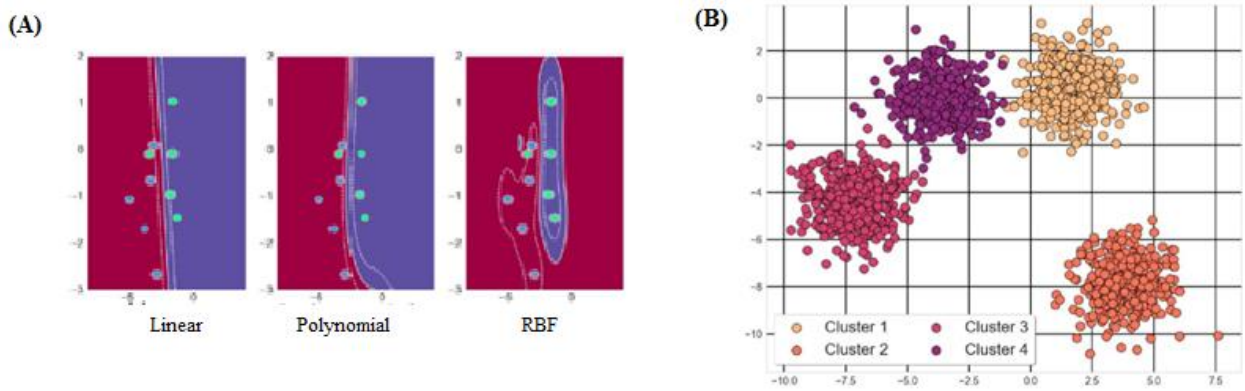
mempertimbangkan komponen "spasial" dari data sedemikian rupa sehingga yang lebih dekat atau lebih dekat secara spasial satu sama lain dalam ruang n-dimensi dikelompokkan bersama (Wunsch et al., (2005)). Tujuan clustering dalam penelitian ini adalah untuk menghasilkan beberapa penjelasan terukur dalam dataset dengan mengatur dan mengelompokkan subset bersama-sama, ini sering digunakan sebagai metode deskriptif sebelum prediksi atau klasifikasi (Mehmed et al., (2011); Wunsch et al., (2005)).

Perpecahan tambahan dalam definisi machine learning hadir dengan pendekatan untuk pembelajaran terawasi dan tidak terawasi. Pembelajaran yang diawasi adalah proses di mana sampel data dimasukkan ke metode dengan label kelas atau output yang diharapkan. Untuk setiap metode pembelajaran yang diawasi, tujuannya adalah membiarkan metode tersebut berjalan, menerapkan metode apa pun yang tersedia untuknya, dan dengan pengetahuannya saat ini, cobalah untuk menebak tentang hasil yang diharapkan. Hasil atau kumpulan hasil dari tebakan ini kemudian diukur terhadap output yang sebenarnya, fungsi internal atau sering kali parameter model diperbarui dengan cermat, dan proses dimulai lagi. Ini pada dasarnya adalah bagaimana sebagian besar machine learning dan metode deep learning baru-baru ini mencapai hasil tertinggi mereka. Pembelajaran tanpa pengawasan seperti pengelompokan mendekati masalah secara berbeda karena aplikasinya umumnya tidak sama dengan pembelajaran yang diawasi. Mereka tidak memiliki label untuk diukur setelah iterasi atau bagian dari pembelajaran. Itu tidak berarti pengelompokan hanya tanpa pengawasan, bila diterapkan dengan benar teknik pengelompokan adalah salah satu yang berkinerja terbaik untuk klasifikasi dalam banyak kasus (Alexandru et al., (2006)). Metode tanpa pengawasan biasanya memiliki tujuan untuk membuat beberapa deskripsi formal tentang data. Contohnya adalah Restricted Boltzmann Machines (RBM) yang mempelajari distribusi probabilitas dari dataset dan algoritme Apriori untuk analisis keranjang pasar. Untuk kelengkapan, ada juga pembelajaran semi-supervised yang mengambil unsur-unsur dari pembelajaran baik yang diawasi maupun yang tidak diawasi (Zhu, (2005)).

Metode pembelajaran

Salah satu teknik machine learning yang berakar pada teori pembelajaran statistik adalah Support Vector Machines (SVM) (Vladimir et al., (1995)). SVM adalah metode pembelajaran terawasi yang dalam bentuk dasarnya adalah pengklasifikasi linier yang memisahkan dua kelas satu sama lain melalui hyperplane. Sebuah hyperplane didefinisikan berdasarkan dot product dari input vektor. Inilah salah satu alasan mengapa metode Quantum Support Vector Machine (QSVM) menjadi populer di bidang QML (Lloyd et al., (2014)). Beberapa hyperplane ada di antara kelas sehingga SVM juga berupaya memaksimalkan jarak antar kelas. Hyperplane dengan jarak maksimal ini ada ketika jaraknya paling jauh dari sampel terdekat dari kedua (semua) kelas. Margin juga merupakan komponen penting untuk SVM karena memberikan kemampuan untuk berkompromi ketika data tidak dapat dipisahkan dengan sempurna seperti pada kebanyakan kasus. Margin adalah ruang batas, berisi hyperplane, antara kelas-kelas tetapi dengan dukungan tambahan untuk memungkinkan tumpang tindih antar kelas. Optimalisasi hyperplane ini dilakukan dengan menggunakan transformasi Lagrangian dalam banyak kasus. Mesin vektor dukungan telah diperluas untuk menyertakan pengklasifikasi nonlinier berdasarkan apa yang disebut trik/metode/fungsi kernel. Mungkin metode kernel yang paling populer adalah fungsi basis radial atau RBF. Metode kernel ini menggantikan produk titik menjadi generalisasi SVM nonlinear yang lebih kuat (Mehmed et al., (2011)). Pada Gambar 1, tiga metode kernel ini di plot: LINEAR, POLYNOMIAL, dan RBF. SVM sangat bergantung pada fungsi-fungsi ini dan memilih yang terbaik didorong oleh data. Contoh berisi sepuluh sampel, lima untuk setiap kelas. Pada gambar, garis putih solid adalah hyperplanes pemisah, dan garis putih putus-putus adalah marginnya.

p-ISSN : 2808-909X e-ISSN : 2798-2505



Gambar 1. (A) Penerapan metode kernel SVM pada sepuluh sampel yang menunjukkan batas, margin, dan hyperplane yang berbeda untuk dataset yang sama. RBF adalah satu-satunya kernel yang hampir mampu mengklasifikasikan semua 10 sampel dengan benar. (B) adalah Empat kelas yang dikelompokkan oleh algoritma KMEANs. Dataset sederhana di sini menunjukkan kemampuan metode penambangan deskriptif pada data cembung

Untuk clustering ada beberapa teknik yang masuk dalam kategori seperti hirarki metode, metode partisi, dan metode berbasis kepadatan. Salah satu pengelompokan yang paling terkenalmetode adalah metode partisi yang disebut algoritma KMEANs (James(1967)). Metode ini agak langsung karena mencoba mengelompokkan data ke dalam kelompok *k* dengan varians yang sama dengan mengurangi *inertia* atau jumlah kuadrat dalam-cluster. Algoritme mengharuskan pengguna memilih nilai *k* yang sering ditemukan baik dengan coba-coba, pendapat ahli, label kelas, atau kombinasi dari semuanya. Pendekatan naif dilakukan dengan menetapkan cluster berdasarkan k dan kemudian memperbarui centroids atau pusat cluster dengan menghitung jarak Euclidean kuadrat terkecil dari sampel. Persamaan Visualisasi metode pengelompokan juga membuatnya menarik saat mencoba menjelaskan metode atau mendapatkan intuisi. Gambar 2 menunjukkan hasil algoritma KMEANs sederhana dengan sampel dua dimensi dengan empat cluster. Hasil ini secara visual dikelompokkan dengan baik dan mudah dapat dibedakan dibandingkan dengan dataset lainnya.

Jalur deep learning diaspal melalui perceptron atau multilayer perceptron (MLP) (Yoshua et al., (2016); Hinton et al., (2015)). Pada intinya, perceptron adalah fungsi sederhana yang mengambil input vektor dan mengambil produk titiknya dengan bobot bernilai nyata. Dalam kasus klasifikasi biner, nilai output akan menjadi satu ketika produk titik lebih besar dari nol dan nol sebaliknya. Sebuah perceptron sangat sederhana dan karena itu tidak dapat menyelesaikan masalah nonlinier (Yoshua et al., (2016)). Grafik pada Gambar 3. menunjukkan bahwa tidak ada satu pun hyperplane yang dapat memisahkan segitiga merah dari oranye titik. Contoh ini menunjukkan bahwa bahkan gerbang logika XOR sederhana pun tidak dapat dipisahkan secara linear, untuk mengatasi hal ini maka beberapa perceptron yang ditumpuk bersama dalam bentuk "lapisan" dapat ditambahkan, beberapa lapisan (biasanya tiga atau lebih) membuat MLP. MLP memungkinkan pendekatan nonlinier untuk dipelajari. Sebuah MLP menghubungkan setiap nodenya (neuron) dengan semua node lainnya dan disebut dengan fully connected layer atau lapisan padat (Mehmed et al., (2011); Yoshua et al., (2016) ; Hinton et al., (2015)).

Beberapa "lapisan" MLP adalah dasar dari deep learning, biasanya berada pada kedalaman beberapa lapisan. Dengan representasi ini maka dimilikilah gagasan tentang lapisan tersembunyi antara lapisan input dan output jaringan. MLP yang digunakan menggunakan dua lapisan padat 64 dan 32 neuron dan satu neuron pada lapisan output, visualisasinya dapat dilihat pada Gambar 4. Output layer dari perceptron dapat dimodifikasi secara ekstensif dengan paket perangkat lunak. Itu tidak hanya dapat melakukan klasifikasi biner tetapi klasifikasi multikelas atau regresi nilai kontinu. Perilaku ini dikendalikan oleh *activation function* (George, (1989)). Dalam gambar 3 (B) Model yang digunakan dalam eksperimen TFQ menggunakan jaringan serupa dengan jumlah node empat kali lipat pada dua layer pertama.

Optimasi dan Rugi

p-ISSN : 2808-909X e-ISSN : 2798-2505

Jika optimasi adalah metode untuk belajar maka metode belajar dapat diartikan sebagai heuristik. Heuristik diberikan kendali atas jenis parameter khusus yang disebut hyperparameter. Dalam SVM ini mungkin toleransi untuk margin atau dalam kasus deep learning jumlah neuron. Awalnya orang yang merancang atau mengimplementasikan suatu teknik akan mengatur hyperparameter ini yang tidak dapat diubah selama pelatihan. Hyperparameter ini digunakan untuk mengembangkan pembelajar karena menentukan cara menghitung dan mengoptimalkan model yang diterapkan (David, (1996)). Ini sering diputuskan menggunakan heuristik tambahan, coba-coba, atau kombinasi keduanya. Pelajar menggunakan parameter ini di dalam model untuk menentukan atau menghitung parameter modelnya sendiri yang tidak terlihat oleh pengguna. Parameter model tidak imajiner atau konseptual. Dalam algoritme sederhana, nilai yang akan dihasilkan parameter ini ditunjukkan secara tepat; namun, karena model berskala ke dataset yang lebih besar dan ruang fitur yang lebih kompleks, ukuran dan jumlah parameter ini cenderung menuju ledakan kombinatorial (Monro et al., (1951)).

Salah satu optimasi yang sering diterapkan dalam deep learning adalah Stochastic Gradient Descent (SGD). SGD adalah metode iteratif yang mencoba melakukan konvergensi pada beberapa minima (lokal atau minimal) di dalam sebuah fungsi yang sesuai dengan data (Monro et al., (1951)). Ini memiliki dua parameter utama untuk menghitung gradien suatu fungsi: w atau bobot dan η yang merupakan laju pembelajaran atau ukuran langkah. SGD adalah fungsi terdiferensiasi yang berbentuk penjumlahan gradien. Dalam prosesnya, SGD menghasilkan bobot baru setelah setiap iterasi. SGD harus bisa menjadi besar sehingga bisa keluar dari minimum yang "buruk" dan cukup kecil untuk tidak memantul dari minimum yang "baik" (Yoshua et al., (2016)). Pengoptimal menerapkan parameter tambahan untuk mengontrol perilaku pembelajaran seperti momentum untuk menyelesaikan masalah yang dihadapi SGD (Yuan et al., (2015)). Secara konseptual, sebuah "panduan" diimplementasikan dalam bentuk fungsi kerugian untuk mengarahkan pengoptimal menuju minima lokal ini (Hinton et al., (2010)). Secara umum, kerugian menentukan bagaimana heuristik telah dilakukan sejauh ini. Dengan menerapkan kerugian ini selanjutnya akan dapat menentukan bagaimana pembelajaran meningkatkan atau menurunkan hasil secara keseluruhan. Ada beberapa fungsi aktivasi yang dapat diterapkan pada neuron. Tiga pilihan yang sangat populer ditunjukkan pada Tabel 2., baik persamaan maupun grafik ditampilkan. Dari ketiganya, Rectified Linear Unit atau ReLU telah membuat gebrakan di komunitas deep learning untuk menunjukkan bahwa ia mampu melakukan konvergensi lebih cepat (Hinton et al., (2010)). Setelah setiap neuron memiliki aktivasi yang diterapkan padanya, maka fungsi kerugian dapat diterapkan ke lapisan atau seluruh jaringan. Pada akhir setiap instance pelatihan atau lebih tepatnya sepasang instance pelatihan, penurunan kerugian menandakan peningkatan kinerja, biasanya akurasi, ini menyiratkan bahwa heuristik telah mempelajari beberapa komponen data (Salakhutdinovet al., (2014)). Saat bagian Eksperimen dikembangkan, fungsi kerugian juga dapat menyiratkan bahwa pelatihan telah berubah menjadi lebih buruk (overfit).

Overfitting adalah keadaan ketika fungsi pembelajaran telah dioptimalkan secara berlebihan atau dalam arti menghafal data pelatihan yang dirahasiakan. Saat melakukan overfitting, pengoptimal berhenti menyesuaikan informasi fitur dan mulai mengoptimalkan untuk mengurangi fungsi kerugian pada sampel pelatihan. Cukup menambahkan dataset validasi untuk digunakan dalam model tidak akan memperbaiki situasi karena masalahnya terletak pada metode atau penerapan metode itu sendiri. Ada beberapa metode untuk mengelola model yang overfits. Masalah utama overfitting dalam penelitian ini terjadi ketika data telah di-fit sebaik mungkin, atau model telah dilatih terlalu lama. Masalah tersebut memiliki beberapa solusi seperti mengurangi kompleksitas metode pembelajaran, mengurangi waktu pelatihan, dan/atau melupakan beberapa informasi yang dipelajari dalam suatu zaman. Lebih baik lagi, seperti yang ditunjukkan kuantum TensorFlow, menggunakan teknik persiapan keadaan kuantum yang salah akan menyebabkan *underfitting* (Mehmed et al., (2011)).

Komputasi Kuantum

Komputasi kuantum memiliki histori yang lebih panjang daripada yang diyakini banyak orang (Chuang et al., (2002); Jozsa et al., (1992)). Peter, (1994) menjelaskan bahwa minat Komputasi Kuantum diaduk dalam komunitas ilmiah. Algoritma Pemfaktoran Shor menunjukkan bahwa teknik kriptanalisis dapat dipercepat secara eksponensial, membahayakan metode yang digunakan untuk melindungi data yang tersimpan dan komunikasi baik sipil dan aplikasi pemerintah. Menurut John, (1969) diantara tantangan teknis yang terkait dengan implementasi fisik komputer kuantum salah satu tantangan ini adalah dekoherensi yang mana gerbang kuantum harus lebih cepat dari hilangnya informasi terhadap lingkungan. Fenomena ini, memaksakan kendala pada komputasi, membuat beberapa algoritma tidak

p-ISSN : 2808-909X e-ISSN : 2798-2505

praktis. Meskipun kebisingan skala besar gratis komputer kuantum tampaknya di luar cakrawala, ada beberapa algoritma kuantum yang bisa dieksekusi dengan teknologi saat ini (Ashley, (2016)).

Penelitian ini dilakukan pada perangkat Kuantum Skala Menengah Bising (NISQ), menghasilkan algoritma dan simulasi untuk teknologi saat ini dan status pengembangan perangkat keras. Komputer klasik tidak dapat melacak dan mendeskripsikan qubit dalam simulasi kuantum dan pada komputer kuantum adalah sebaliknya, itu dapat dilakukan dengan mudah. Postulat Mekanika Kuantum bersifat aljabar, artinya ada hubungan intrinsik antara komputasi kuantum dan operasi aljabar. Berbagai kemajuan di bidang pemrosesan informasi kuantum telah memberikan prospek yang menjanjikan dengan mengandalkan keunggulan itu. Oleh karena itu, telah terbukti bahwa Quantum Computing dapat mempercepat eksponensial dalam pemrosesan data yang berbeda dan metode machine learning, termasuk Principal Component Machines (PCA) (Paul et al., (1987); Rebentrost et al., (2014)), K-means Clustering (Tang et al., (2010)) dan Mesin Vektor Dukungan yang teratur (SVM) (Lloyd et al., (2014)). Kemajuan perangkat NISQ menyiratkan pengembangan aplikasi yang lebih beragam dan bermakna, meningkatkan relevansi melakukan penelitian di bidang ini.

Kebutuhan komputasi kuantum berasal dari keinginan untuk memodelkan dunia nyata dengan detail. Dalam kasus yang paling mendasar, beberapa qubit secara komputasi sulit untuk disimulasikan dan dilacak oleh perangkat klasik. Itu tidak berarti perangkat klasik akan usang ketika / jika perangkat kuantum mencapai supremasi atas mereka. Perangkat kuantum dapat dilihat sebagai unit komputasi sekunder di luar Computing Processing Unit (CPU) biasa yang menangani perhitungan dengan kompleksitas yang jauh lebih besar seperti Graphics Processing Unit (GPU). Salah satu cara perangkat kuantum mampu melakukan kalkulasi sulit yang tidak bisa dilakukan perangkat klasik adalah dengan memanfaatkan ruang Hilbert. Ruang Hilbert adalah generalisasi ruang vektor dengan struktur hasil kali dalam, merupakan ruang lengkap (huang et al., (2002)). Ruang Hilbert adalah ruang hasil kali dalam yang nyata atau kompleks yang juga merupakan ruang metrik lengkap sehubungan dengan fungsi jarak yang diinduksi oleh hasil kali dalam.

Komputer kuantum saat ini berbentuk perangkat Noisy Intermediate Scale Quantum (NISQ). Perangkat ini sama sekali bukan keadaan akhir komputer kuantum, tetapi batu loncatan untuk mempersiapkan, mengembangkan, dan menguji algoritme. Penskalaan perangkat ini menimbulkan beberapa tantangan, meskipun itu bukan tujuan dari penelitian ini, tetapi harus disebutkan. Sistem fisik rentan terhadap kesalahan karena masalah seperti kebisingan stokastik. Penelitian di bidang ini disebut sebagai koreksi kesalahan kuantum dan berfungsi untuk meningkatkan kesetiaan sistem kuantum dalam keadaan yang tidak ideal ini (John, (1969); Negrevergne et al., (2001)). Perangkat komputasi kuantum ketika dibuat menggunakan proses litograf memiliki koneksi fisik antara qubit yang disebut persimpangan Josephson (Alexander et al., (1999)). Persimpangan Josephson adalah persimpangan terowongan yang terdiri dari dua logam superkonduktor yang dipisahkan oleh penghalang isolasi. Fenomena tersebut merupakan produk dari penerowongan kuantum (Chuang et al., (2002)). Perangkat kuantum dengan karakteristik ini termasuk perangkat IBM seperti pada Gambar 7. Perangkat ini memiliki jumlah qubit yang berbeda tetapi tidak secara langsung berhubungan dengan kemampuan komputasinya. IBM telah menjuluki istilah volume kuantum untuk menunjukkan kapasitas komputer kuantum dengan mempertimbangkan beberapa faktor seperti jumlah qubit, kedalaman sirkuit sebelum tingkat kesalahan terlalu besar, konektivitas topologi, crosstalk, kesalahan gerbang U, CNOT kesalahan, antara lain.

Data dapat diubah untuk mewakili keadaan kuantum menggunakan perubahan sewenang-wenang apa pun ke basis komputasi qubit. Dataset sederhana seperti $\{x, y, z\}$ dapat menerapkan gerbang kesatuan atau gerbang U. Sebuah unitary dioperasikan atas sekumpulan input yang menghasilkan sekumpulan output untuk mendapatkan sekumpulan keadaan transformasi baru. Konsep atau skema untuk menerapkan kesatuan pada data diberikan pada Gambar 8.

Machine learning Kuantum

Istilah quantum machine learning (QML) dapat memiliki lebih dari satu arti tergantung pada penggunaannya. Tujuan dari persilangan ini adalah untuk mengetahui apakah penambahan komponen kuantum dapat dimanfaatkan untuk meningkatkan pembelajaran metode klasik. Model QML yang akan diterapkan dalam penelitian ini dilakukan pada perangkat kuantum yang disimulasikan yakni machine learning berbantuan kuantum. Jenis pembelajaran ini dapat dibagi menjadi beberapa pengaturan yang berbeda tetapi mungkin contoh yang paling jelas diberikan dalam sebagai peta empat kuadran. Peta pada Gambar 9. mewakili empat jenis pendekatan machine learning berbantuan kuantum yang dapat

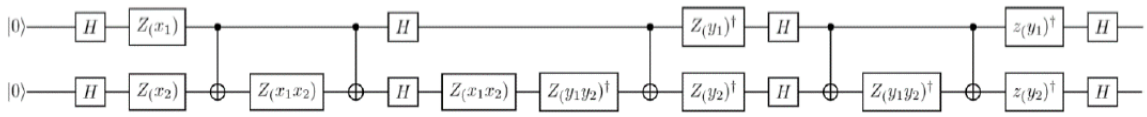
p-ISSN : 2808-909X e-ISSN : 2798-2505

diambil. Komponen tersebut merupakan gabungan dari dua huruf dimana yang pertama adalah pola dasar data dan yang kedua adalah perangkat komputasi. Huruf pertama adalah data kuantum "Q" atau "C" klasik dan huruf kedua adalah perangkat "Q" kuantum atau "C" klasik. Oleh karena itu, kombinasi QC hanya pada bagian ini merupakan singkatan yang berarti data kuantum pada perangkat klasik. Sistem "Q" pada gambar 8 (B) adalah kuantum dan C adalah klasik. Karakter pertama mewakili sumber data dan yang kedua mewakili perangkat. CQ dibaca data klasik pada perangkat kuantum.

Komputasi dalam QML sebagian besar telah berkembang ke keadaan di mana ada beberapa model kerja yang telah dikembangkan dan dipelihara secara teoritis dalam "Kebun Binatang" kuantum online (Ashley, (2016)). Metode ini berbagi persilangan dari model klasik standar dan dalam beberapa kasus hanyalah adaptasi data kuantum dalam model klasik. Dua dari model ini diimplementasikan di sini, yaitu Quantum Convolutional Neural Network (QCNN) dan Variation Quantum Classifier (VQC) (Ashley, (2016)). Secara umum, algoritma pembelajaran yang diawasi dalam penelitian ini dibahas karena QCNN akan dikembangkan di bagian Pengaturan Eksperimental TFQ, terutama model VQC. Kedua model ini dicirikan sebagai optimasi kuantum parametrik. Konsep percobaan berulang adalah metode penanganan noise pada perangkat era NISQ. Tembakan dilakukan sedemikian rupa sehingga distribusi probabilitas hasil dapat dipastikan. Nilai dalam distribusi hasil dengan nilai terbesar dianggap sebagai output nilai sebenarnya dari suatu percobaan. Penerapan bidikan dalam model QML diterapkan per iterasi model. Dengan cara ini jika dilakukan dua puluh iterasi pengoptimalan dengan ukuran batch lima dan sepuluh tembakan, maka totalnya adalah 1.000 eksperimen pada perangkat kuantum. Ini menunjukkan keadaan QML pada perangkat era NISQ. Jumlah bidikan yang diperlukan bergantung pada kerumitan rangkaian. Pada gambar 10 eksperimen di implementasikan dan menunjukkan rangkaian yang mencari nilai biner 1111. Rangkaian tersebut merupakan variasi empat qubit dari algoritme Grover yang membutuhkan 632 gerbang. Jumlah percobaan yang dilakukan pada kedua kasus adalah 8.192 dan nilai 15 adalah hasil dari sistem kira-kira 1/3 dari waktu.

2. TINJAUAN PUSTAKA

Secara umum, penelitian ini menghadirkan peningkatan praktis dan solusi untuk pengklasifikasi parametrik pada perangkat era NISQ. Para penulis juga menunjukkan bahwa metode mereka dapat digabungkan dalam model klasik-kuantum yang menggunakan ResNet, kumpulan bobot model backbone deep learning, untuk model Quantum Approximate Optimization Algorithm (QAOA). Pendekatan pertama adalah rangkaian kuantum variasional yang menerapkan pengukuran biner. Pendekatan kedua mengikuti dari SVM klasik yang memanfaatkan konstruksi hyperplanes untuk memperkirakan metode kernel. Eksperimen ini dilakukan pada prosesor kuantum lima qubit dari IBM. Mungkin komponen yang paling mencolok dari pekerjaan ini adalah metode kernel yang sangat nonlinier yang harus dibangun agar akurasi tinggi dapat dicapai. Rangkaian tersebut mampu mencapai akurasi 100% pada dataset yang dihasilkan menggunakan rangkaian yang ditunjukkan pada Gambar 11 dan merupakan salah satu dari beberapa upaya untuk memecahkan masalah serupa. Metode mereka mengacu pada gagasan bidikan karena kebisingan dan kemampuan perangkat era NISQ saat ini.



Gambar 11. Sirkuit kuantum Havlíček untuk mengklasifikasikan dataset kecil yang sangat nonlinier.

Pemrograman dan Kerangka Kerja

Paket komputasi kuantum merupakan dasar untuk mengembangkan teknik QML. Memanfaatkan paket-paket ini untuk QML membutuhkan pengetahuan tentang komputasi kuantum dan machine learning. Pustaka komputasi kuantum berisi banyak komponen atau fungsi dasar untuk membuat gerbang dasar, menggunakan simulator, mengukur hasil, membuat oracle, dan lain-lain. Pustaka QML di sisi lain menyediakan seperangkat alat berbeda yang dapat digunakan di atas paket QC. Pustaka ini seperti yang ditunjukkan menggunakan elemen dari machine learning di atas pustaka komputasi kuantum

TensorFlow Quantum

TensorFlow Quantum (TFQ) diumumkan oleh Google pada awal tahun 2020 sebagai pustaka baru untuk machine learning kuantum. Implementasinya dan dukungan saat ini hanya untuk Bahasa Python. TFQ

p-ISSN : 2808-909X e-ISSN : 2798-2505

dibuat dapat diakses publik, pedoman untuk mengembangkan, menguji, dan merancang model sederhana disediakan dalam dokumentasinya. Pustaka TFQ dibangun di atas basis TensorFlow (TF) yang telah menjadi pemimpin terkenal untuk pengembangan deep learning. TFQ bermaksud untuk menyediakan prototipe cepat dari model machine learning klasik kuantum hibrida. Pustaka TFQ bekerja sama dengan dua pustaka lain untuk matematika simbolik (Scopatz et al., (2017)) dan desain sirkuit logika kuantum yang masing-masing Sympy dan Cirq. Pustaka ini sangat mendasar untuk membuat model pembelajaran di TFQ. Karena TFQ membutuhkan paket-paket lain ini untuk melakukan QML, mereka juga harus dikembangkan sebagai bagian dari tumpukan perangkat lunak untuk percobaan.

Cirq adalah paket desain sirkuit yang menyediakan beberapa kemampuan yang sama seperti QASM dan Qiskit Aer. Cirq beberapa tahun lebih tua, dirilis pada 2018, dari TFQ. Ini dikembangkan oleh Tim Quantum AI Google yang pada intinya adalah komponen tumpukan perangkat lunak yang memungkinkan komputasi kuantum. Cirq dimaksudkan untuk dapat digunakan pada simulator lokal mesin pengguna. Saat ia melakukan operasi kuantum universal, jika ditranspilasikan dengan benar, ia dapat menjadi agnostik perangkat saat digunakan pada perangkat kuantum yang sebenarnya. Sympy adalah perpustakaan untuk matematika simbolik dan merupakan Computer Algebra System (CAS) berfitur lengkap dengan histori yang lebih panjang daripada perpustakaan komputasi kuantum baru-baru ini. Sympy dimaksudkan untuk dimanfaatkan oleh mereka yang membutuhkan perhitungan matematika yang benar. Kegunaannya antara lain Kalkulus, Matematika Diskrit, Geometri, Fisika, Kombinatorika (Scopatz et al., (2017)). Sympy awalnya dirilis pada tahun 2006 dan bukan merupakan hasil dari usaha Google ke dalam ruang kuantum. Penggunaan Sympy dalam penelitian ini untuk mengontrol parameter dalam model kuantum. Mereka berbeda dari parameter tipikal karena pustaka TensorFlow dibangun di atas dua komponen yaitu placeholder dan grafik TensorFlow untuk komputasi model deep learning. Sympy digunakan dalam grafik sebagai placeholder variabel kuantum parametrik untuk nilai menengah yang disediakan oleh perhitungan kuantum dalam model.

Paket Non-Quantum

Coding, eksperimentasi, dan pengembangan penelitian ini dilakukan di Python3 untuk porsi pekerjaan di TFQ menggunakan Python 3.6.10. Anaconda adalah program yang memungkinkan manajemen paket sederhana dan kontrol lingkungan, digunakan untuk membuat lingkungan terpisah untuk TFQ. Sebagian besar paket diinstal menggunakan saluran 'anaconda' utama atau 'conda-forge', ketika kedua saluran ini tidak memiliki paket khusus, Package Installer for Python (PIP) digunakan. Sebagai Integrated Development Environments saat mengembangkan kode maka digunakanlah Visual Studio Code dan Jupyter Notebook. Numpy adalah pustaka untuk operasi aljabar linier dan vektor/matriks, yang digunakan secara luas baik dalam menerapkan beberapa langkah prapemrosesan dan pasca analisis. Scikit-Learn digunakan untuk membuat dataset, memisahkan data, dan melakukan banyak langkah pasca analisis. Pustaka Scikit-Learn adalah pustaka besar dengan sub modul untuk dataset yang tidak seimbang, pemrosesan gambar, dan banyak tugas penambangan data. Ini adalah perpustakaan yang umumnya berputar di sekitar metode pembelajaran umum, persiapan data untuk metode pembelajaran, dan analitik prediktif. Itu dibangun menggunakan Matplotlib, Numpy, dan Scipy (John et al., (2007); Olivier et al., (2011)). Scikit-Learn juga digunakan dalam langkah pasca analisis untuk membuat kurva ROC/AUC dan mengumpulkan metrik hasil pembelajaran. Beberapa langkah prapemrosesan yang dijelaskan di bagian Eksperimen dilakukan dengan menggunakan pustaka Scikit-Learn. Pustaka Pandas digunakan untuk memanipulasi dan melihat data melalui DataFrames yang mempermudah penanganan dan transformasi data. DataFrames juga memungkinkan fungsi diterapkan kolom dengan bijaksana membuat banyak langkah rumit menjadi mudah.

Visualisasi adalah komponen kunci untuk memfasilitasi pemahaman banyak transformasi. Itu juga merupakan media utama utama untuk mengekspresikan langkah-langkah preprocessing, dan hasil dari penelitian ini. Matplotlib digunakan untuk sebagian besar plot dan sebagian besar terkait dengan Numpy baik dalam aplikasi praktis maupun pengembangan internal (John et al., (2007)). Ini menyediakan fungsi untuk plot seperti pencar, garis, histogram, kepadatan, diagram lingkaran, antara lain. Seaborn juga digunakan untuk memplot, ia memperluas Matplotlib dengan menambahkan gaya dan beberapa fungsi plot tambahan saat menggunakan Pandas DataFrames. Plotting Poincare dilakukan dengan menggunakan Plotly, plot ini digunakan dalam analisis parameter Stokes. Plotly adalah perpustakaan visualisasi multibahasa dengan kemampuan merencanakan yang luas seperti Matplotlib. Untuk menyimpan hasil dan data, maka file Comma Separate Value (CSV) dan JavaScript Object Notation (JSON) digunakan. CSV membuat tampilan data menjadi sangat sederhana saat dibagikan dan dievaluasi satu per satu. Baik CSV dan JSON

p-ISSN : 2808-909X e-ISSN : 2798-2505

mudah digunakan dengan Python dan memiliki pustaka bawaan untuk menangani keduanya. Penulisan dilakukan dengan menggunakan Microsoft Word dan editor Overleaf online untuk LaTeX. Microsoft Visio diguanakn untuk membuat grafik unik dari plot data.

Dataset

Dalam menjelajahi dataset dengan metode kuantum, sejumlah distribusi dan bentuk yang berbeda akan diuji. Dataset yang digunakan sebagian besar adalah dataset sintetis dengan alasan, data sintetik adalah untuk menunjukkan betapa berbedanya perilaku metode pembelajaran kuantum pada mereka. Bentuk data saat mengembangkan dataset untuk menjawab hipotesis juga akan dikontrol sehingga mampu menghasilkan beberapa dataset menggunakan metode Scikit-Learn Generator. Selain itu, dataset 'mainan' yang merupakan utilitas sebelum menguji metode pada dataset sebenarnya juga akan digunakan. Dataset 'mainan' yang digunakan adalah dataset Iris dan dataset Wine

Dataset Generator Scikit-Learn

Empat dataset dibuat menggunakan Scikit-Learn. Dua dataset dari dataset scikit-learn Toy dimodifikasi dari desain out of the box agar sesuai dengan analisis penelitian. Dataset di kelas Generator Sickit-Learn memiliki beberapa parameter yang dapat diatur agar sesuai dengan tujuan yang ingin dicapai. Fungsi generator di Scikit-Learn memiliki parameter yang dapat dikontrol untuk jumlah fitur, jumlah sampel, status acak (untuk reproduktifitas), jumlah nilai berulang, dan parameter unik untuk jenis data yang dapat dihasilkan oleh fungsi tersebut. Parameter ini akan mengontrol pemisahan antar kelas dalam dataset.

Membuat Dataset Blobs

Di Generator MakeBlobs, parameter CenterBox akan menentukan penyebaran setiap sampel di dalam kelas. Ketika jumlah kelas hanya dua, parameter CenterBox menjadi centroid dari setiap kelas sepanjang sumbu y positif dan negatif. Dalam dua dimensi, parameter CenterBox jika sama dengan (-4.5, 4.5) akan menghasilkan kelas satu berpusat di sekitar -4.5 dan kelas dua berpusat di sekitar 4.5 keduanya sepanjang sumbu y. Ketika jumlah kelas lebih besar dari dua maka CenterBox tidak lagi menjadi centroid tetapi menjadi kotak pembatas untuk setiap pusat cluster. Dalam kedua kasus, rentang yang lebih besar dalam parameter CenterBox menyiratkan sampel yang lebih terdistribusi secara kompak per kelas dengan umumnya lebih sedikit tumpang tindih, sedangkan rentang yang lebih kecil menyiratkan lebih banyak tumpang tindih antar kelas dan lebih sedikit kekompakan. Nilai untuk CenterBox dapat berkisar dari (-10, 10). Contoh dua dimensi dari data MakeBlobs ditunjukkan pada Gambar 12 CenterBox adalah (-3, 3). Generator MakeBlobs digunakan untuk algoritma KMEANs di Pendahuluan.

Membuat Dataset Circles

Dalam Generator MakeCircles parameter faktor menentukan faktor ruang antara dua lingkaran konsentris. Dataset hanya memiliki dua fitur sehingga ketika digunakan untuk tiga - atau empat dimensi data fitur ketiga dan keempat dapat dihasilkan dari distribusi normal, melalui padding dengan konstanta, atau keduanya. Dengan faktor 0,9 lingkaran dalam akan sangat dekat dengan lingkaran luar lingkaran hampir tumpang tindih itu. Jika faktornya kecil seperti 0,2 lingkaran dalam akan jauh lebih kecil dan memiliki lebih banyak jarak antara itu dan lingkaran luar. Generator MakeCircles saja menghasilkan dua kelas output. Kebisingan juga dapat ditambahkan ke kedua lingkaran dalam bentuk standar penyimpangan untuk distribusi Gaussian diterapkan saat membuat lingkaran. Contoh dari a Sampel MakeCircles yang dihasilkan ditunjukkan pada Gambar 12(B) mengandung noise sangat sedikit dan faktor yang sangat kecil.

Membuat Dataset Moons

Di Generator MakeMoons tidak ada parameter tambahan untuk mengontrol bentuk atau lokasi kedua kelas tersebut. Generator ini membuat dua setengah bulan atau busur di mana salah satu ujungnya masing-masing "bulan" kelas berada di puncak yang lain. Dataset hanya memiliki dua fitur jadi ketika digunakan untuk data tiga atau empat dimensi fitur ketiga dan keempat dapat dihasilkan dari normal distribusi, melalui padding dengan konstanta, atau keduanya. Kebisingan dapat ditambahkan ke kedua bulan dalam bentuk dari standar deviasi untuk distribusi Gaussian yang diterapkan saat membuat bulan. Sebuah visual dari MakeMoons ditunjukkan pada Gambar 14, ini mengandung sangat sedikit noise.

Membuat Dataset Swiss Role

Di Generator MakeSwissRole ada sejumlah parameter yang dapat dikontrol untuk dibentuk dataset output. Detail yang paling penting adalah bahwa tidak ada label kelas yang mendefinisikan komponen dataset. Dataset SwissRole perlu diklasifikasikan dengan metode yang berbeda untuk menentukan kelas per sampel

p-ISSN : 2808-909X e-ISSN : 2798-2505

data. Metode yang diterapkan mengikuti langsung dari dokumentasi sebagai metode yang paling dapat direproduksi, tetapi harus dinyatakan bahwa kemungkinan besar bukanlah satu-satunya metode terbaik. Ini menambah lapisan kerumitan saat pengelompokan metode untuk menghasilkan label kelas ditentukan untuk dataset dan kemudian mengharapkan model QML untuk dikenali konten dari data saat memisahkan kelas. Seluruh dataset dapat dilihat pada Gambar 15. setelah itu dilakukan clustering menggunakan metode Agglomerative Clustering untuk menghasilkan label kelas. Metode Agglomerative menghasilkan total enam kelas ketika dikelompokkan pada dataset.

Dataset Iris

Dataset Iris, penambangan data, atau analitik sampai tingkat tertentu. Asal dataset dapat ditemukan dari repositori publik database di situs web UCI untuk Machine Learning. Dataset Iris dianggap dalam banyak kerangka kerja sebagai aplikasi dasar alat pada serangkaian langkah "nyata" untuk menerapkan model. Ini terutama karena mudah untuk mencapai hasil yang sangat tinggi untuk model klasifikasi dengan data ini. Itu juga hanya berisi empat fitur dan tiga kelas output yang sesuai dengan jenis bunga Iris. Empat fitur Panjang Petal, Lebar Petal, Panjang Sepal, dan Lebar Sepal sesuai dengan sifat fisik bunga. Ketiga kelas tersebut adalah Setosa, Versicolour, dan Virginica. Tujuan saat menggunakan. Dataset Iris adalah untuk menentukan kelas bunga mana yang diwakili fitur tersebut. Dataset ini digunakan secara ekstensif untuk mengembangkan beberapa kesimpulan. Dataset Iris Ddiplot pada Gambar 16. dalam dua bentuk. Yang pertama adalah distribusi nilai untuk setiap fitur (sepanjang diagonal) dan plot pencar dari setiap pasangan fitur juga ditampilkan. Harap perhatikan bahwa segitiga bawah dan atas matriks berisi plot pencar yang sama, keduanya disertakan untuk preferensi tampilan.

Dataset Wine

Dataset Wine adalah dataset mainan publik lainnya yang dapat ditemukan di dalam Scikit-Learn atau situs web UCI untuk ML. Data Wine berisi 14 fitur, dan tujuannya adalah menggunakan fitur ini untuk mengklasifikasikan salah satu dari tiga jenis wine. Kelas-kelas ini diberikan sebagai (0,1,2). Matriks korelasi diberikan per kelas pada Gambar 17. Menunjukkan hubungan antara setiap dua fitur dalam data. Matriks korelasi menunjukkan terutama bahwa untuk kelas 0 fitur sebagian besar berkorelasi negatif sedangkan fitur di kelas 2 sebagian besar berkorelasi positif. Dataset hanya berisi nilai positif yang berkelanjutan. Fitur termasuk alkohol, malic_acid, ash, flavonoid, color_intensity, dan rona untuk beberapa nama. Dataset ini awalnya dimaksudkan untuk digunakan dalam model TFQ.

3. METODOLOGI PENELITIAN

Secara umum eksperimen ini menggunakan TensorFlow Quantum (TFQ). Eksperimendari TFQ bertujuan untuk membangun model klasik kuantum hibrida yang mampu melampaui hasil model dokumentasi TFQ untuk dataset MakeBlobs. Hal ini dilakukan dengan melakukan beberapa perubahan pada encoding data yaitu dengan menerapkan Amplitude Encoding. Setelah mengembangkan dan membangun argumen untuk Pengkodean Amplitudo, analisis data yang lebih konkret dan langkah-langkah preprocessing dilakukan dan beberapa langkah ini diulangi lagi dari bagian TFQ. Pekerjaan di TFQ relatif terbatas pada analisis Pengkodean Amplitudo di bagian Transformasi untuk Pembelajaran Dalam Model Kuantum karena dikembangkan sepenuhnya melalui eksperimen TFQ.

Persiapan Keadaan Kuantum

Seperti yang ditunjukkan persiapan keadaan kuantum adalah faktor penting agar teknik Quantum Machine Learning (QML) berhasil. Langkah-langkah ini diterapkan dalam fase preprocessing sebagai sarana untuk mengkodekan informasi sebelum melakukan pembelajaran apa pun. Pada bagian ini pertama-tama tiga metode persiapan keadaan dikembangkan: pengkodean basis, pengkodean sudut, dan pengkodean amplitudo yang merupakan metode untuk mempersiapkan keadaan kuantum dari data klasik.

Pengkodean Sudut

Pengkodean sudut adalah metode yang sederhana dan efektif untuk pengkodean informasi, tetapi tidak kuat, dan tidak memetakan informasi dari keadaan klasik dengan cara yang terdefinisi dengan baik. Pengkodean sudut pada dasarnya adalah bentuk paling dasar dari pengkodean data klasik ke dalam keadaan kuantum. Ini memiliki hasil yang baik untuk masalah seperti pemeriksaan paritas atau bekerja dengan rentang nilai tertentu yang terbatas yang sebagian besar tidak dapat diterapkan pada dataset nyata. Pengkodean sudut dilakukan dengan menerapkan rotasi gerbang terhadap sumbu x $R_x(v_i)$ atau sumbu y $R_y(v_i)$ di mana v_i

p-ISSN : 2808-909X e-ISSN : 2798-2505

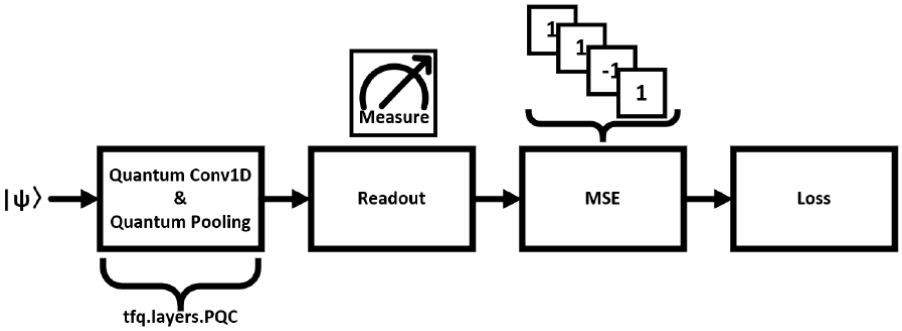
adalah nilai yang akan dikodekan. Dalam ruang Hilbert, rotasi terhadap sumbu y menerapkan rotasi sudut, biasanya berdasarkan beberapa π, oleh karena itu disebut pengodean sudut. Pertimbangkan dataset klasik dengan tiga fitur di mana satu record diwakili oleh vektor v = ([0.1], [0.2], [0.3]). Dalam rotasi sudut jumlah rotasi yang diterapkan akan sama dengan jumlah fitur dalam dataset, R_x diterapkan pada v tiga kali satu kali untuk setiap dimensi. Sampel yang dihasilkan dalam bentuk gerbangnya ditunjukkan pada Gambar 18. di mana $|q1\rangle$, $|q2\rangle$, $|q3\rangle$ adalah qubit yang akan mengambil status baru $|\psi_1\rangle$, $|\psi_2\rangle$, $|\psi_3\rangle$ mewakili nilai vektor yang dikodekan v_1, v_2, v_3. Sampel n-dimensi akan membutuhkan n jumlah qubit untuk menghasilkan himpunan keadaan kuantum. Metode ini membuat representasi sederhana dari data dengan kompleksitas yang kira-kira sama seperti yang terjadi pada representasi klasiknya. Hal ini membuat pengkodean sudut menarik untuk dataset sederhana yang mungkin memiliki sedikit perbedaan di antara sampel. Perangkat era NISQ memiliki jumlah qubit yang terbatas dan menjaga lebih dari beberapa koheren untuk eksperimen itu sulit.

Pengkodean Amplitudo

Pengkodean amplitudo memetakan data klasik ke dalam amplitudo qubit. Secara konseptual itu dapat dianggap sebagai pengkodean lain yang harus mewakili tetapi juga tanpa kehilangan dapat dikodekan. Perbedaan dengan pengkodean amplitudo adalah *basis komputasi* untuk memungkinkan anggapan keadaan berbeda *dari* pengkodean basis dan sudut. Encoding one-hot mengubah bentuk sampel dari padat menjadi jarang. Baik dalam kasus klasik atau kuantum, langkah preprocessing ini dapat berdampak besar pada kinerja metode pembelajaran.

Penyiapan Eksperimental TFQ

Meskipun eksperimen di bagian ini dilakukan di perangkat klasik yang mensimulasikan komputer kuantum, metodologi dan penyiapannya sama. Ini dilakukan karena saat ini tidak ada komputer kuantum yang tersedia secara publik dari Google untuk TFQ. Dua set percobaan dilakukan berdasarkan jumlah masa pelatihan, dengan delapan zaman (percobaan satu) dan lima puluh zaman (percobaan dua). Tugasnya adalah mengklasifikasikan dua kelas -1 dan +1. Untuk mengevaluasi perilaku pelatihan pengkodean amplitudo mengarah ke model yang akan menyatu pada akurasi yang lebih tinggi lebih cepat (i) dan pengkodean amplitudo mengarah pada pelatihan yang tidak menentu (ii) dan sehingga akan lebih efektif dari waktu ke waktu. Komponen klasik adalah MLP, komponen kuantum model adalah Quantum Convolution Neural Network (QCNN). Arsitektur model QCNN yang digunakan ditunjukkan pada Gambar 19 adalah inti dari model yang dapat ditemukan dalam dokumentasi TFQ dan juga digunakan dalam analisis tetapi dibiarkan tidak berubah sebagai garis dasar. Model ini terdiri dari sejumlah konvolusi kuantum satu dimensi (QConv1D), dengan lapisan kumpulan kuantum (QPool) pada QConv1D. Pembacaan keadaan kuantum dilakukan setelah rangkaian dilakukan dengan menerapkan gerbang Z pada qubit setelah lapisan terakhir penyatuan. Sirkuit untuk QConv1D ditunjukkan di bagian atas Gambar 20. Lapisan QPool adalah non-parametrik, lapisan ini hanya menerapkan gerbang Pauli X, Y, dan Z ke sirkuit.



Gambar 19. Arsitektur sederhana diimplementasikan untuk setiap model TFQ. Ini adalah keseluruhan arsitektur untuk model QCNN yang dikembangkan di sini.

Eksperimen TFQ menggunakan tiga model, dua di antaranya mengikuti dokumentasi TFQ untuk Quantum Convolutional Neural Networks (QCNN) dan yang ketiga adalah desain penelitian ini sendiri. Model TFQ (a) QCNN sebagai model dasar yang berisi lapisan kuantum fundamental di ketiga model, (b) Angle-Hybrid yang menerapkan pengkodean sudut untuk persiapan keadaan, dan (c) Amplitudo-Hybrid yang menerapkan pengkodean amplitudo untuk keadaan persiapan. Dua yang terakhir adalah model hybrid sehingga akan berisi QCNN dan MLP secara berurutan. Metodologi yang diterapkan pada dataset

p-ISSN : 2808-909X e-ISSN : 2798-2505

MakeBlobs yang ditunjukkan pada Gambar 21. Secara total, untuk delapan dan lima puluh zaman dengan delapan dataset. Setiap dataset diberikan dalam plot dua dimensi untuk menunjukkan kisaran kesulitan berdasarkan centroid. Data dimulai dengan centroid 0,6 di mana kedua kelas sebagian besar tumpang tindih dan diakhiri dengan centroid 2,0 di mana kedua kelas jauh lebih dapat dipisahkan. Masing-masing dataset berisi empat fitur dan dan membagi data menjadi 2.048 sampel untuk pelatihan, 512 untuk validasi dalam model, dan 512 untuk pengujian atau evaluasi setelah pelatihan. Selanjutnya parameter fungsi model dan perhitungan metrik yang diterapkan dalam analisis menunjukkan kehilangan, akurasi, presisi, daya ingat, dan skor F1 hasilnya dihitung dan ditampilkan. Banyak fungsi model yang sama diterapkan dari dokumentasi TFQ.

Analisis Eksperimen TFQ – Model Hibrida

Secara umum, hipotesis penelitian ini adalah: menjalankan model untuk sejumlah zaman pendek dengan pengkodean amplitudo akan menghasilkan hasil pembelajaran yang lebih baik. Ini juga akan menyiratkan bahwa waktu pelatihan yang lebih sedikit diperlukan untuk mencapai hasil yang lebih baik. Nilai (-1) dan (+1) dalam dua tabel di subbagian ini mengacu pada metrik individual label kelas. Set tes kedua membiarkan model berjalan selama lima puluh zaman untuk mendapatkan sampel yang representatif dari histori pembelajaran. Analisis utama berasal dari kombinasi plot pelatihan dan tabel. Model QCNN sebagian besar disertakan untuk referensi saat melakukan analisis.

Eksperimen: Delapan zaman

Hasil validasi pelatihan diberikan pada Tabel 3. Hasil ini menunjukkan Amplitudo-Hybrid adalah yang berkinerja terbaik di setiap metrik evaluasi. Untuk dua centroid pertama, 0,6 dan 0,8, dataset yang paling sulit dari percobaan ini, model Amplitudo-Hybrid mencapai hasil akurasi sekitar 2% lebih baik daripada model Angle-Hybrid. Dapat dilihat bahwa jarak 1,4 centroid model Amplitudo-Hybrid mencapai akurasi 90%. Sebaliknya, Angle-Hybrid mencapai akurasi 90% hanya pada jarak 2,0 centroid. Untuk setiap model meningkat saat jarak pusat massa menyebar lebih jauh atau saat bergerak ke bawah melalui tabel. Untuk Angle-Hybrid dengan jarak 1,4 centroid dapat dilihat bahwa hampir setiap metrik, tidak termasuk kerugian, untuk setiap kelas berada di atas 90%. Plot (gambar 22) menunjukkan akurasi validasi pelatihan dan kehilangan setiap dataset per setiap model (QCNN, Angle-Hybrid, dan Amplitude-Hybrid). Dengan Angle-Hybrid perilaku pembelajaran model datar tidak mengaitkan optimasi dan pembelajaran yang stabil. Selain jarak centroid 1,8 dan 2,0 model tampaknya belajar dengan cepat dan konsisten selama delapan zaman. Hasil ini memberikan gambaran bahwa model yang menggunakan pengkodean amplitudo dalam beberapa zaman jauh lebih unggul.

Rata-rata hasil Tabel 3. menunjukkan beberapa hasil tambahan, ini juga mendukung Amplitudo-Hybrid. Model terbaik di sini ditentukan lagi dengan melihat akurasi dan hasil kelas gabungan per metrik. Dengan hasil QCNN dan Angle-Hybrid menunjukkan bahwa mereka serupa dalam banyak kasus. Hasil ini tidak mengherankan karena mereka menggunakan metode pengkodean sudut yang sama dan karena itu hasilnya berkorelasi erat untuk setiap kesulitan. Oleh karena itu, dengan atau tanpa penambahan komponen klasik/hibrid, ada sedikit peningkatan akurasi selama delapan zaman. Dalam kasus terbaik, Angle-Hybrid hanya memiliki akurasi dua persen lebih baik daripada model QCNN tetapi dalam sebagian besar kasus akurasinya berada dalam 0,5% untuk kedua model ini. Oleh karena itu, model yang menggunakan pengkodean sudut membatasi kemampuan pembelajaran secara keseluruhan dan dari Tabel 3. komponen tambahan model Angle-Hybrid tidak meningkatkan kinerja. Pada percobaan selanjutnya terdapat kesamaan yang mengingatkan pada hasil tersebut.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Model	Centriod	Loss	Accuracy	Precision (-1)	Recall (-1)	F1-Score (-1)	Precision (1)	Recall (1)	F1-Score (1)
QCNN	0.6	0.829	0.683	0.755	0.564	0.646	0.639	0.808	0.713
Angle-Hybrid	0.6	0.965	0.687	0.729	0.618	0.669	0.655	0.760	0.703
Amplitude-Hybrid	0.6	0.889	0.708	0.767	0.618	0.684	0.667	0.804	0.729
QCNN	0.8	0.739	0.726	0.796	0.625	0.700	0.679	0.832	0.748
Angle-Hybrid	0.8	0.808	0.740	0.779	0.687	0.730	0.708	0.796	0.749
Amplitude-Hybrid	0.8	0.702	0.763	0.819	0.690	0.749	0.721	0.840	0.776
QCNN	1.0	0.660	0.767	0.832	0.683	0.750	0.720	0.856	0.782
Angle-Hybrid	1.0	0.672	0.781	0.807	0.751	0.778	0.757	0.812	0.783
Amplitude-Hybrid	1.0	0.533	0.826	0.864	0.782	0.821	0.792	0.872	0.830
QCNN	1.2	0.595	0.792	0.848	0.725	0.781	0.750	0.864	0.802
Angle-Hybrid	1.2	0.563	0.800	0.812	0.793	0.803	0.789	0.808	0.798
Amplitude-Hybrid	1.2	0.396	0.865	0.897	0.832	0.863	0.836	0.900	0.867
QCNN	1.4	0.544	0.826	0.864	0.782	0.821	0.792	0.872	0.830
Angle-Hybrid	1.4	0.475	0.833	0.836	0.839	0.838	0.831	0.828	0.829
Amplitude-Hybrid	1.4	0.281	0.902	0.930	0.874	0.901	0.875	0.932	0.903
QCNN	1.6	0.507	0.859	0.874	0.847	0.860	0.844	0.872	0.858
Angle-Hybrid	1.6	0.406	0.853	0.845	0.874	0.859	0.863	0.832	0.847
Amplitude-Hybrid	1.6	0.183	0.939	0.949	0.931	0.940	0.929	0.948	0.938
QCNN	1.8	0.484	0.871	0.862	0.889	0.875	0.880	0.852	0.865
Angle-Hybrid	1.8	0.357	0.876	0.861	0.904	0.882	0.894	0.848	0.870
Amplitude-Hybrid	1.8	0.113	0.966	0.965	0.969	0.967	0.967	0.964	0.965
QCNN	2.0	0.473	0.892	0.881	0.912	0.896	0.904	0.872	0.887
Angle-Hybrid	2.0	0.286	0.902	0.878	0.938	0.907	0.931	0.864	0.896
Amplitude-Hybrid	2.0	0.101	0.974	0.966	0.984	0.975	0.983	0.964	0.973

Table 3. Eight-Epoch Post Analysis Model Metrics.

Model	Centriod	Loss	Accuracy	Precision (-1)	Recall (-1)	F1-Score (-1)	Precision (1)	Recall (1)	F1-Score (1)
QCNN	0.6	0.798	0.687	0.729	0.618	0.669	0.655	0.760	0.703
Angle-Hybrid	0.6	1.013	0.683	0.771	0.541	0.636	0.634	0.832	0.719
Amplitude-Hybrid	0.6	1.171	0.685	0.798	0.515	0.626	0.629	0.864	0.728
QCNN	0.8	0.708	0.753	0.785	0.713	0.748	0.726	0.796	0.759
Angle-Hybrid	0.8	0.818	0.742	0.821	0.633	0.715	0.690	0.856	0.764
Amplitude-Hybrid	0.8	0.953	0.718	0.835	0.561	0.671	0.657	0.884	0.754
QCNN	1.0	0.631	0.787	0.814	0.755	0.784	0.762	0.820	0.789
Angle-Hybrid	1.0	0.641	0.796	0.859	0.721	0.784	0.750	0.876	0.808
Amplitude-Hybrid	1.0	0.727	0.796	0.898	0.679	0.773	0.732	0.920	0.815
QCNN	1.2	0.568	0.830	0.832	0.835	0.834	0.827	0.824	0.825
Angle-Hybrid	1.2	0.497	0.839	0.891	0.782	0.833	0.797	0.900	0.845
Amplitude-Hybrid	1.2	0.522	0.849	0.930	0.763	0.838	0.791	0.940	0.859
QCNN	1.4	0.520	0.849	0.843	0.866	0.854	0.855	0.832	0.843
Angle-Hybrid	1.4	0.384	0.865	0.903	0.824	0.862	0.831	0.908	0.868
Amplitude-Hybrid	1.4	0.367	0.896	0.948	0.843	0.892	0.853	0.952	0.899
QCNN	1.6	0.486	0.876	0.864	0.900	0.882	0.891	0.852	0.871
Angle-Hybrid	1.6	0.300	0.896	0.916	0.877	0.896	0.877	0.916	0.896
Amplitude-Hybrid	1.6	0.260	0.925	0.970	0.881	0.924	0.886	0.972	0.927
QCNN	1.8	0.467	0.888	0.875	0.912	0.893	0.903	0.864	0.883
Angle-Hybrid	1.8	0.242	0.917	0.926	0.912	0.919	0.909	0.924	0.916
Amplitude-Hybrid	1.8	0.160	0.949	0.975	0.923	0.949	0.924	0.976	0.949
QCNN	2.0	0.347	0.902	0.889	0.923	0.906	0.916	0.880	0.897
Angle-Hybrid	2.0	0.300	0.894	0.906	0.885	0.895	0.882	0.904	0.893
Amplitude-Hybrid	2.0	0.115	0.970	0.976	0.965	0.971	0.964	0.976	0.970

Tabel 4. Metrik Model Analisis Pasca Lima Puluh Zaman.

Eksperimen: Lima puluh zaman

Untuk hasil disini, bentuk output yang sama di plot pada Gambar 23 dan Tabel 4 tetapi kali ini model diizinkan berjalan selama lima puluh zaman. Konsistensi selama periode pembelajaran dan skenario apa pun yang menonjol adalah tanda bahaya. Pertama, selama delapan periode pelatihan, di mana hal-hal secara umum tampak meningkat di tempat yang tepat untuk setiap model. Kecenderungan ini sebagian besar berlanjut lagi tetapi jelas bahwa beberapa model berkinerja lebih buruk daripada setelah lima puluh zaman. Dengan membandingkan tabel ke 3 dan Tabel 4 maka beberapa kasus tentang Angle-Hybrid secara keseluruhan akan ditampilkan. Pelatihan sewenang-wenang untuk jumlah zaman yang lebih besar bukanlah cara yang efektif untuk mencapai peningkatan akurasi. Dalam kisaran epoch 10-20 setiap model telah menyesuaikan diri dengan data sebaik mungkin, dengan akurasi puncak, penurunan kerugian. Munculnya overfitting menyiratkan model akan mulai mempertahankan banyak pengaruh dari data pelatihan. Overfitting di sini menandakan model telah melakukan semua yang bisa dilakukan untuk mempelajari informasi yang ada dalam data. Sampai taraf tertentu perilaku menandakan model belajar dari data dalam perilaku yang diharapkan secara. Secara keseluruhan, ini menyiratkan perilaku pembelajaran terbaik yang dapat dicapai adalah dengan menggunakan pengkodean Amplitudo dan konsisten.

Dengan model QCNN dan Angle-Hybrid perbedaan hasil historis dari zaman ke zaman dapat dilihat (Gambar 23). Hasil untuk keduanya menunjukkan bahwa model ini tidak pernah cocok dengan data. Dalam

p-ISSN : 2808-909X e-ISSN : 2798-2505

beberapa kasus, model membuat peningkatan yang baik pada metrik keseluruhannya. Namun, dalam beberapa kasus beberapa hasil melatih dengan baik selama dua lusin zaman pertama dan kemudian muncul lagi untuk meningkatkan setelah dua lusin kemudian. Kunci dari analisis ini adalah penampilan ini membaik. Melihat kembali pada wilayah zaman 10-20, jelas kesimpulan sebelumnya untuk Amplitudo-Hybrid tidak dapat ditarik di sini. Model Angle-Hybrid di wilayah ini melonjak dari akurasi "tinggi" dan kemudian dalam beberapa zaman turun 5% menjadi 15%. Ini terjadi tidak hanya di sini tetapi selama seluruh periode pelatihan. Perilaku ini menyiratkan bahwa pembelajaran gagal karena heuristik mencoba untuk "menebak" solusi yang lebih baik daripada yang dilakukan pada iterasi sebelumnya. Dengan meninjau dua model hybrid untuk jarak centroid 1.0, 1.2, dan 1.4 dapat dilihat bahwa perbedaan akurasi yang mendukung Amplitudo-Hybrid adalah 0,0%, 0,976%, dan 3,125% masing-masing. Tidak hanya akurasi, tetapi juga metrik lain seperti skor F1 menunjukkan ada sedikit perbedaan antara model-model ini. Recall dan Presisi model Angle-Hybrid sebenarnya lebih baik di sini daripada Amplitudo-Hybrid. Meskipun demikian, informasi pada Tabel 4 menyesatkan. Selama periode pelatihan untuk dataset ini setiap model Angle-Hybrid memantulkan 5-10% epoch ke epoch sementara model Amplitudo-Hybrid sedikit berubah untuk dataset ini. Sekali lagi, melihat 1.2 lebih spesifik dalam hal Gambar 23, model Amplitudo memuncak pada 14 zaman tetap di sana selama beberapa zaman dan perlahan-lahan menurun selama sisa pelatihan. Saat mencari perilaku serupa di Amplitudo-Hybrid, dapat dilihat bahwa hal yang sama terjadi untuk setiap dataset lainnya. Namun, hal ini tidak terlihat pada model Angle-Hybrid. Jika dilihat dari Tabel 4, maka dapat disimpulkan bahwa hasil tabel tidak sekuat karena overfitting. Hal yang sama tidak berlaku untuk model Angle-Hybrid.

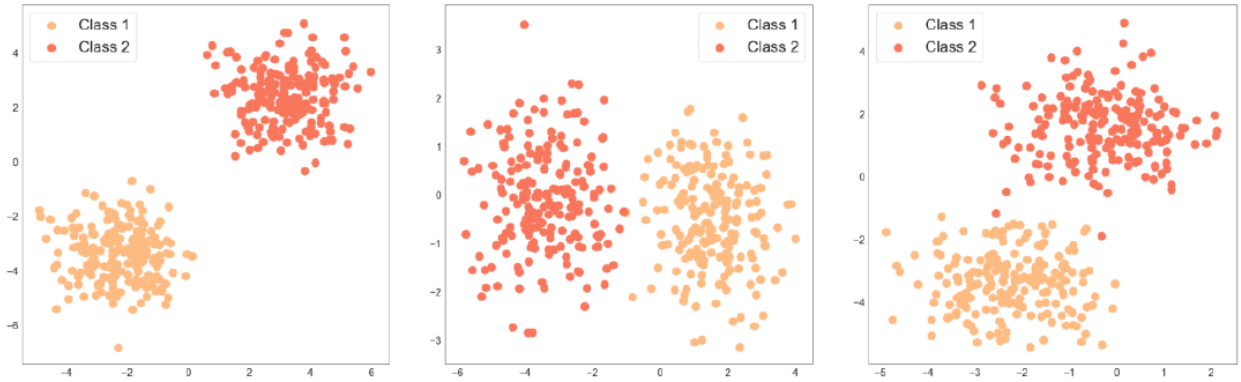
Transformasi untuk Pembelajaran dalam Model Kuantum

Proses eksperimen bagian ini dilakukan menggunakan Variational Quantum Classifier (VQC) untuk mengevaluasi bagaimana langkah-langkah prapemrosesan tertentu dapat memengaruhi pelatihan algoritme kuantum penuh. Setiap metode yang dapat diterapkan ke kumpulan tidak dapat dicakup semua sehingga beberapa lainnya perlu diuji dan dievaluasi menggunakan sumber data yang berbeda. Sebagian besar metode di sini menyimpang dari penyetelan hyperparameter apa pun karena itu tidak sama dengan data prapemrosesan. Dalam penelitian ini, sebagian besar metode dikembangkan menggunakan manipulasi array Numpy sederhana atau fungsi Scikit-Learn paket. Setiap dataset dibuat dengan ukuran sampel 400 poin dan status acak 11.

Analisis Pembuatan Dataset Blobs

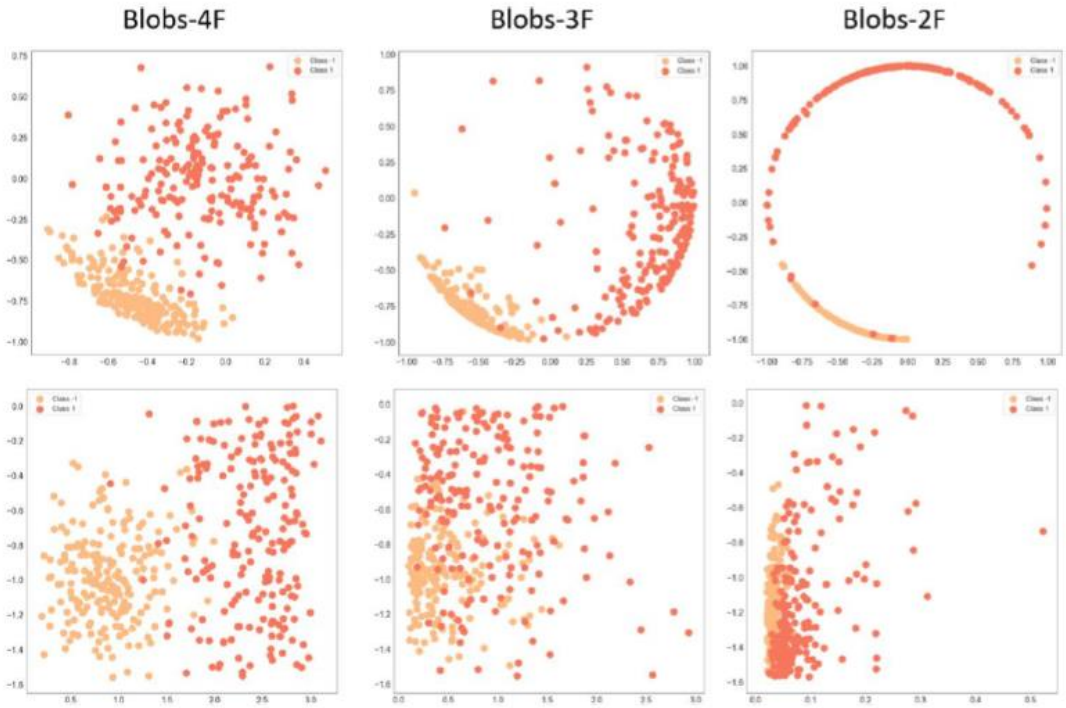
Generator MakeBlobs memiliki kemampuan untuk mengontrol beberapa parameter saat membuat dataset. Kemampuan ini digunakan dalam pembuatan tiga dataset yakni Blobs-4F, Blobs-3F, dan Blobs-2F. Ketiga dataset dibuat dengan tujuan melakukan klasifikasi biner hanya dengan dua fitur. Penjelasan ketiga dataset blob berikut ini diberikan dengan catatan bahwa pembaca dapat mengimplementasikan solusi serupa dengan menggunakan metode yang berbeda. Dalam Blobs-4F mendefinisikan jumlah fitur menjadi empat dan jumlah kelas output target (berpusat pada daftar parameter fungsi) sebagai tiga, dan mengatur CenterBox sama dengan (-3.5, 3.5). Parameter ini menghasilkan distribusi fitur yang berbeda jika dibandingkan dengan dua kelas dan dua fitur. Distribusi yang berbeda inilah yang akan ditangkap menggunakan fitur-fitur ini. Plot titik-titik di bawah pada Gambar 24 berisi tiga dataset ini. Ini menunjukkan bagaimana titik-titik tersebar secara berbeda dari intuisi yang pada awalnya akan memandu pengguna fungsi generator untuk percaya. Parameter ketiga dari CenterBox dipilih karena cukup sederhana sehingga pengklasifikasi tidak perlu merumuskan solusi yang sangat non-linear dalam menentukan batas keputusan dari dua kelas.

p-ISSN : 2808-909X e-ISSN : 2798-2505



Gambar 24. Dataset tiga blob (dari kiri ke kanan adalah Blobs-4F, Blobs-3F, dan Blobs-2F) sebelum pemrosesan awal

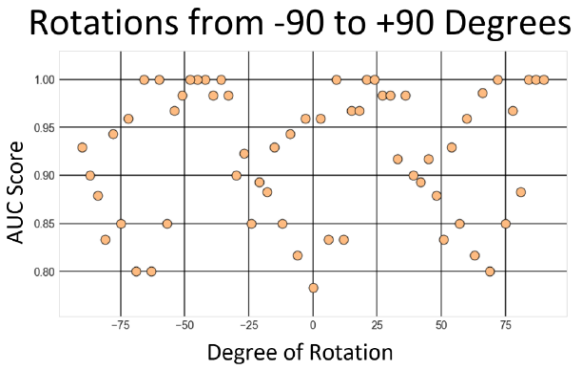
Perubahan ruang koordinat memang meningkatkan kemampuan belajar. Pertimbangkan batas keputusan untuk garis dasar, tambahan dua lapisan, dan di sini dengan rotasi sekitar -90 derajat. Dengan lapisan tambahan, pengklasifikasi mencoba menghasilkan garis vertikal meskipun mengklasifikasikan kira-kira lima puluh persen sampel secara tidak benar. Hasil saat berputar adalah garis vertikal lagi tetapi jauh lebih akurat setelah rotasi. Dalam kasus dua yang pertama, hasil yang serupa seharusnya dicapai dengan beberapa garis horizontal. Rotasi ini menimbulkan pertanyaan tentang rotasi lain, mana yang meningkatkan, dan mana yang selanjutnya menghambat pengklasifikasi? Penginputan koefisien ke fitur rotasi tidak menunjukkan peningkatan luar biasa dengan rotasi sehingga analisis rotasi dikecualikan. Dalam dataset Wine memiliki beberapa tantangan menarik, masalah utamanya adalah fitur mana yang terbaik untuk perangkat kuantum atau lebih umum fitur mana yang cukup dapat dipisahkan untuk dipelajari. Dalam hal klasifikasi biner, masalah ini tidak sulit, dan mudah untuk menunjukkan bahwa kelas pertama dan ketiga berbeda dalam banyak fiturnya. Hal ini dapat dilihat pada Gambar 17. Beberapa fitur telah dilatih dan diterapkan pada langkah-langkah dasar untuk persiapan status dan dari dataset Iris. Beberapa fitur yang dapat dipisahkan "kurang" dan pelatihan data ini berkinerja buruk dipilih dengan harapan dapat ditingkatkan baik melalui penerapan rotasi atau metode penskalaan/normalisasi yang berbeda. Analisis lebih lanjut di sini perlu dilakukan pada masing-masing himpunan bagian yang digunakan dengan derajat yang berbeda-beda.



p-ISSN : 2808-909X e-ISSN : 2798-2505

Gambar 26. Gumpalan dataset setelah menerapkan MinMaxScalar dan padding ke empat fitur (atas) dan vektor fitur setelah menerapkan metode translasi sudut ke data (bawah)

Berbagai faktor yang diterapkan pada data setelah normalisasi awal juga diuji. Faktor-faktor ini diterapkan pada dua dataset, Wine dan MakeSwissRolls. Mengalikan satu fitur dengan sepuluh dan fitur lainnya dengan lima akan menyebabkan penurunan performa untuk Iris, Wine, MakeBlobs, dan MakeSwissRolls. Analisis tambahan harus diterapkan untuk menentukan apakah faktor sesuai dengan distribusi nilai dan pengujian nilai interval harus dievaluasi. Parameter Stokes juga diterapkan setelah padding dan kemudian menghapus fitur ketiga dari analisis. Parameter Stokes telah dimanfaatkan dalam metodologi QML dan telah menunjukkan hasil yang dapat diterapkan untuk prapemrosesan. Dalam hal efektivitas, ini diterapkan pada MakeBlobs, MakeCircles, dan MakeMoons. Untuk MakeBlobs, penerapan parameter Stokes adalah performa terbaik untuk Blobs-4F, tetapi performanya lebih buruk daripada baseline di semua kasus lainnya. Untuk parameter MakeCircles Stokes bekerja hanya ketika data sangat terpisah yang memberikan sedikit wawasan. Untuk MakeCircles, hasilnya buruk tetapi sedikit lebih buruk daripada model dasar. Metode pada sample dua dimensi pada penelitian ini sebagian besar sudah diuji dan dikembangkan, ini juga diperluas ke dataset 3, 4 dan 5 dimensi. Kesimpulan dari eksperimen dengan sampel berdimensi lebih tinggi yang lebih besar adalah bahwa kesederhanaan model VQC menghambat apa pun selain tingkat pengoptimalan yang dangkal di semua dataset yang diuji. VQC dicoba diterapkan pada data diatas 2dimensi tetapi penyetelan hyperparameter dasar dan beberapa langkah prapemrosesan yang berbeda tidak berhasil. Kesimpulan ini meskipun tidak berguna dalam hal kuantifikasi menunjukkan bahwa model VQC tidak benar-benar cocok untuk ruang fitur yang kompleks. Secara lebih umum, ada beberapa model VQC dan pengklasifikasi kuantum parametrik lainnya seperti QSVM dan QAOA yang mampu menangani spektrum masalah. Sampai batas tertentu, VQC yang diterapkan pada Iris menipu dan memiliki sedikit penerapan komputasi di luar diskusi yang telah dikembangkan. Akar dari hal ini berasal dari preprocessing yang diterapkan pada dataset Iris. Ada sedikit keraguan bagi penulis bahwa data disiapkan sesuai dengan metode yang diterapkan.



Gambar 27. Skor AUC dari 61 model yang dilatih berdasarkan rotasi dataset. Rotasi dilakukan setiap tiga derajat. Perilaku periodik yang jelas dapat dilihat dari hasil kira-kira setiap lima puluh derajat.

Transformasi terakhir adalah penerapan transformasi yang biasanya dilakukan setelah parameter Stokes. Plot data ini untuk dua dataset ditunjukkan pada Gambar 28, meskipun diterapkan pada setiap dataset Generator. Metode preprocessing ini disebut "Transformasi Poincare" karena memplot data ke lingkup Poincare tanpa menerapkan parameter Stokes terlebih dahulu. Dua dataset tersebut awalnya hanya memiliki dua fitur. Untuk setiap generator yang memiliki dua fitur maka dalam fitur ketiga akan diisi dengan dua cara berbeda: dengan konstanta atau sekumpulan konstanta dan dengan beberapa distribusi nilai. Distribusi dan konstanta divariasikan untuk menemukan sampel menarik yang mungkin berkinerja baik atau tidak dalam model kuantum. Dataset pada Gambar 28 adalah yang paling menjanjikan dari dataset yang dihasilkan ini. Di antara dataset yang menjanjikan ini juga ada MakeGaussianQuantiles, MakeSCurve, dan MakeSwissRolls (untuk pasangan kelas tertentu). Dua di antaranya, MakeGaussianQuantiles dan MakeSCurve yang diuji dalam transformasi Poincare ini tidak seperti dataset lainnya. Metode normalisasi yang berbeda juga diterapkan pada beberapa data Poincare. Citra pada Gambar 28 menggunakan normalisasi L-1, L-2, dan L-Maximum. Normalisasi yang berbeda ini diujikan pada Iris, MakeBlobs, MakeMoons, dan MakeSwissRoll. Dari jumlah tersebut, Iris adalah satu-satunya dataset yang diklasifikasikan dengan akurasi di atas 90%. Dataset lainnya bekerja lebih baik saat menerapkan MinMaxScalar ke data.

p-ISSN : 2808-909X e-ISSN : 2798-2505

KESIMPULAN DAN SARAN

Komponen preprocessing dan model QML telah dikerjakan dan diimplementasikan. Pada paruh pertama eksperimen, metode persiapan keadaan (pengkodean amplitudo) sudah diterapkan dan diuji. Hasil seperti tolok ukur untuk dua aplikasi persiapan juga didapatkan. Eksperimen ini dilakukan menggunakan framework TensorFlow Quantum. Bagian kedua dari penelitian ini mencoba menjelaskan beberapa langkah yang membuat pelatihan model kuantum menjadi lebih kuat. Metode-metode ini sebagian dilakukan sebagai analisis untuk memahami apa yang perlu dipertimbangkan untuk mempersiapkan metode di masa mendatang untuk QML. Pada dasarnya, penelitian ini hanya mengembangkan metode biner karena banyak literatur di bidang ini menawarkan hasil dari jenis yang sama. Pekerjaan masa depan di bidang ini akan membutuhkan perluasan ke dalam teknik klasifikasi multikelas yang cukup dikembangkan untuk dapat diterapkan dalam pekerjaan seperti ini.

Meskipun perangkat era NISQ rusak dan memiliki jalan panjang, mereka sesuai untuk menguji dataset dengan fitur jumlah kecil. Pemanfaatan perangkat NISQ pada transformasi dan analisis yang diterapkan dalam penelitian ini akan menjadi langkah selanjutnya. Ada banyak masalah dengan menggunakan perangkat era NISQ modern mulai dari kebisingan dan dekoherensi hingga batasan waktu dan ruang yang diberikan. Hal-hal ini antara lain adalah apa yang mengecilkan penggunaannya sebagai pengganti perangkat simulasi yang lebih banyak tersedia. Simulator kuantum telah berkembang pesat dan telah memberikan hasil yang layak dan hasil yang dapat diterima untuk bergerak maju menggunakan implementasi komputer kuantum yang berisik. Karena kemampuan perangkat kuantum saat ini, hasil secara umum diperkirakan lebih buruk. Harapannya adalah muncul hasil atau pola yang mirip dengan hasil belajar. Metode yang diterapkan pada penelitian ini tidak dijamin akan berperilaku sama tetapi dengan antisipasi mengharapkan kesamaan yang besar.

DAFTAR PUSTAKA

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, OndˇrejˇCertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E.Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, ShivamVats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R.Terrel, Štˇepán Rouˇcka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing inpython. PeerJ Computer Science, 3:e103, January 2017.

Benioff, Paul. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines." Journal of statistical physics 22, no. 5 (1980): 563-591.

Berthiaume, André, and Gilles Brassard. "Oracle quantum computing." Journal of modern optics 41, no. 12 (1994): 2521-2535.

Boneh, Dan, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world." In International Conference on the Theory and Application of Cryptology and Information Security, pp. 41-69. Springer, Berlin, Heidelberg, 2011.

Britt, Keith A., and Travis S. Humble. “High-performance computing with quantum processing units.” ACM Journal on Emerging Technologies in Computing Systems (JETC) 13, no. 3 (2017): 1-13.

Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." In Proceedings of the 23rd international conference on Machine learning, pp. 161-168. 2006.

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20, no. 3 (1995): 273-297.

Cory, David G., Amr F. Fahmy, and Timothy F. Havel. "Ensemble quantum computing by NMR spectroscopy." Proceedings of the National Academy of Sciences 94, no. 5 (1997): 1634-1639.

Cybenko, George. "Approximation by superpositions of a sigmoidal function." Mathematics of control, signals and systems 2, no. 4 (1989): 303-314.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Deutsch, David, and Richard Jozsa. "Rapid solution of problems by quantum computation." Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439, no. 1907 (1992): 553-558.

Ge, Rong, Furong Huang, Chi Jin, and Yang Yuan. "Escaping from saddle points— online stochastic gradient for tensor decomposition." In Conference on Learning Theory, pp. 797-842. 2015.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep learning. Vol. 1, no. 2. Cambridge: MIT press, 2016.

Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in science & engineering 9, no. 3 (2007): 90-95.

Kantardzic, Mehmed. Data mining: concepts, models, methods, and algorithms. John Wiley & Sons, 2011.

Knill, Emanuel, Raymond Laflamme, Rudy Martinez, and Camille Negrevergne. "Benchmarking quantum computers: the five-qubit error correcting code." Physical Review Letters 86, no. 25 (2001): 5811.

Kubat, Miroslav, and Stan Matwin. "Addressing the curse of imbalanced training sets: one-sided selection." In Icml, vol. 97, pp. 179-186. 1997.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521, no. 7553 (2015): 436-444.

Lloyd, Seth, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis." Nature Physics 10, no. 9 (2014): 631-633.

MacKay, David JC. "Hyperparameters: optimize, or integrate out?." In Maximum entropy and bayesian methods, pp. 43-59. Springer, Dordrecht, 1996.

MacQueen, James. "Some methods for classification and analysis of multivariate observations." In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, no. 14, pp. 281-297. 1967.

Makhlin, Yuriy, Gerd Scöhn, and Alexander Shnirman. "Josephson-junction qubits with controlled couplings." nature 398, no. 6725 (1999): 305-307.

Montanaro, Ashley. "Quantum algorithms: an overview." npj Quantum Information 2, no. 1 (2016): 1-8.

Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." In ICML. 2010.

Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." (2002): 558-559.

Patra, Bishnu, Rosario M. Incandela, Jeroen PG Van Dijk, Harald AR Homulle, Lin Song, Mina Shahmohammadi, Robert Bogdan Staszewski et al. "Cryo-CMOS circuits and systems for quantum computing applications." IEEE Journal of Solid- State Circuits 53, no. 1 (2017): 309-321.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

Preskill, John. "Reliable quantum computers." Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, no. 1969 (1998): 385-410.

Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification." Physical review letters 113, no. 13 (2014): 130503.

Rebentrost, Patrick, Masoud Mohseni, and Seth Lloyd. "Quantum support vector machine for big data classification." Physical review letters 113, no. 13 (2014): 130503.

Robbins, Herbert, and Sutton Monro. "A stochastic approximation method." The annals of mathematical statistics (1951): 400-407.

Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." In Proceedings 35th annual symposium on foundations of computer science, pp. 124-134. Ieee, 1994.

p-ISSN : 2808-909X e-ISSN : 2798-2505

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15, no. 1 (2014): 1929- 1958.

Wold, Svante, Kim Esbensen, and Paul Geladi. "Principal component analysis." Chemometrics and intelligent laboratory systems 2, no. 1-3 (1987): 37-52.

Xiao, Jing, YuPing Yan, Jun Zhang, and Yong Tang. "A quantum-inspired genetic algorithm for k-means clustering." Expert Systems with Applications 37, no. 7 (2010): 4966-4973.

Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." IEEE Transactions on neural networks 16, no. 3 (2005): 645-678.

Zeng, William, and Bob Coecke. "Quantum algorithms for compositional natural language processing." arXiv preprint arXiv:1608.01406 (2016).

Zhu, Xiaojin Jerry. Semi-supervised learning literature survey. University of Wisconsin- Madison Department of Computer Sciences, 2005.