STATUS OF THESIS

Title of thesis	ENHANCED HYPER-PARAMETER OF SEMI-SUPERVISED GENERATIVE ADVERSARIAL NETWORKS BASED ON SINE COSINE ALGORITHM FOR MULTIMEDIA DATASETS

I ANAS ABDO SALEH SALEH AL-RAGEHI

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP

2. The IRC of UTP may make copies of the thesis for academic purposes only.

3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

____

____

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

____

____

____

Endorsed by

Ts. Dr. Said Jadid Abdulkadir
Associate Professor,
Department of Computer & Information Sciences,
Universiti Teknologi PETRONAS (UTP)

Signature of Author

Signature of Supervisor

Permanent address: One South . Seri Kembangan, 43300, Selangor, Malaysia

Name of Supervisor
AP.Dr. Said Jadid Abdulkadir

Date: 09-06-2022

Date: 09-06-2022

UNIVERSITI TEKNOLOGI PETRONAS

ENHANCED HYPER-PARAMETER OF SEMI-SUPERVISED GENERATIVE ADVERSARIAL NETWORKS BASED ON SINE COSINE ALGORITHM FOR MULTIMEDIA DATASETS

.

by

ANAS ABDO SALEH SALEH AL-RAGEHI

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Signature: Ts. Dr. Said Jadid Abdulkadir
Associate Professor,
Department of Computer & Information Sciences,
Universiti Teknologi PETRONAS (UTP)

Main Supervisor: AP. Dr. Said Jadid Abdulkadir

Signature: Ts. Dr. ALIZA SARLAN
Chair,
Computer and Information Sciences
Universiti Teknologi PETRONAS

Head of Department: Dr. Aliza Bt Sarlan

Date: 09-06-2022

ENHANCED HYPER-PARAMETER OF SEMI-SUPERVISED GENERATIVE ADVERSARIAL NETWORKS BASED ON SINE COSINE ALGORITHM FOR MULTIMEDIA DATASETS

by

ANAS ABDO SALEH SALEH AL-RAGEHI

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER

COMPUTER SCIENCE AND INFORMATION SYSTEMS
UNIVERSITI TEKNOLOGI PETRONAS
BANDAR SERI ISKANDAR
PERAK

JUNE 2022

DECLARATION OF THESIS

Title of thesis	ENHANCED HYPER-PARAMETER OF SEMI-SUPERVISED GENERATIVE ADVERSARIAL NETWORKS BASED ON SINE COSINE ALGORITHM FOR MULTIMEDIA DATASETS

I ANAS ABDO SALEH SALEH AL-RAGEHI

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Ts. Dr. Said Jadid Abdulkadir
Associate Professor,
Department of Computer & Information Sciences,
Universiti Teknologi PETRONAS (UTP)

Signature of Author	Signature of Supervisor
Permanent address: One South Garden, Seri Kembangan, 43300,Selangor, Malaysia.	Name of Supervisor AP. Dr. Said Jadid Abdulkadir
Date: 09-06-2022	Date: 09-06-2022

CONFIRMATION BY PANEL OF EXAMINERS

I certify that a panel of examiners has met on **25/05/2022** to conduct the final examination of **ANAS ABDO SALEH SALEH AL-RAGEHI**, Matric number: **(18000600)** on his thesis entitled: ENHANCED HYPER-PARAMETER OF SEMI-SUPERVISED GENERATIVE ADVERSARIAL NETWORKS BASED ON SINE COSINE ALGORITHM FOR MULTIMEDIA DATASETS

The panel of examiners recommends that the student be awarded the **MASTER DEGREE**. The members of the panel of examiners were as follows:

AP Ts Dr. Low Tang Jung
Associate Professor: Computer & Information Science
Universiti Teknologi PETRONAS
(Chairman)

AP Ts Dr. Norshakirah Bt A Aziz
Associate Professor: Computer & Information Science
Universiti Teknologi PETRONAS
(Internal Examiner)

AP Dr. AbdulRahman Ahmed Al-Sewari
Associate Professor: Computer & Information Science
Birmingham City University- UK
(External Examiner)

AP Dr. Balbir Singh A/L Mahinder Singh
Dean: Centre for Graduates Studies
Universiti Teknologi PETRONAS
Date: 09-06-2022

DEDICATION

To my lovely parents, siblings and wife.

ACKNOWLEDGEMENTS

I would like to give my special regards to everyone involved in making this research a success. First and foremost, in the Name of Allah, the Merciful, the Beneficent, all praise is due to Allah, the Lord of the Worlds.

Secondly, I’d like to express my gratitude and deepest thanks to my advisor, Dr. Said Jadid Abdulkadir, for the appreciated support, guidance, advice, lessons, and tolerance that he has given me throughout this Master's research journey. Dr. Said has taught me a lot of skills and techniques to achieve this research on the perfect and particular path. This research was quite a challenge for me, and without his guidelines and support, I would not be able to show great progress while working on this research.

More than that, I should also appreciate my brother Dr. Fateh and friends and my friend Dr. Mohammed Abdo Albaom. This research would not have been possible without your help and guidance.

A special thanks to Universiti Teknologi PETRONAS for giving me the opportunity to do my master's degree research in Part-time mode and the Computer & Information Science Department staff for their guidance and evaluation throughout my Symposiums every semester.

Finally, I would like to dedicate my special thanks to my father, mother, wife, and brothers for their encouragement, moral support, inspiration, and motivation during my study in one of the best technological universities, Universiti Teknologi PETRONAS (UTP)

ABSTRACT

Generative Adversarial Networks (GANs) are neural networks that allow models to learn deep representations without requiring a large amount of training data. Semi-Supervised GAN Classifiers are a recent innovation in GANs, in which GANs are used to classify generated images not just into real and fake, but also into multiple classes, similar to a general multi-class classifier. However, GANs have a sophisticated design that can be challenging to train. Obtaining the proper set of parameters for all model generator, discriminator and classifiers are difficult. This is because GAN is sensitive to hyperparameter optimization because the benefit of the cost function can be overridden by the benefit of hyperparameter optimization if the cost function is not properly optimized. As a result, the performance of the cost functions can fluctuate across different hyperparameter settings, which can be depressing for developers who are unsure whether the GAN' model is working or not. Consequently, training a single GAN model for different datasets may not produce satisfactory results. Therefore, the SGAN model (Semi-Supervised GAN Classifier) has been proposed in this study. First, a baseline model is constructed, then the model has been enhanced leveraging the Sine-Cosine Algorithm and Synthetic Minority Oversampling Technique (SMOTE). SMOTE was used to address class imbalances in the dataset, while SCA was used to optimize the weights of the classifier models. The optimal hyper parameters (learning rate and batch size) were obtained using grid manual search. Four well-known benchmark datasets and evaluation measures are used to validate the proposed model. The proposed method was then compared against existing models, and the results on each dataset were recorded and demonstrated the effectiveness of the proposed model. The proposed model successfully improved test accuracy scores by 1%, 2%, 15%, and 5% on MNIST digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection Dataset respectively.

ABSTRAK

Empunyai reka bentuk yang canggih yang boleh mencabar untuk dilatih. Ini kerana untuk mendapatkan set parameter yang betul untuk semua model - penjana, diskriminasi dan pengelas adalah sukar, dan akibatnya, melatih satu model GAN untuk set data yang berbeza mungkin tidak menghasilkan hasil yang memuaskan. Oleh itu, dalam kajian ini model SGAN (Pengkelas GAN Separuh Seliaan) telah dicadangkan. Mula-mula model garis dasar dibina, kemudian model itu telah dipertingkatkan dengan memanfaatkan Algoritma Sinus-Kosinus dan Teknik Pensampelan Lebihan Minoriti Sintetik (SMOTE). SMOTE digunakan untuk menangani ketidakseimbangan kelas dalam set data manakala SCA digunakan untuk mengoptimumkan berat model pengelas. Set hiperparameter optimum (kadar pembelajaran dan saiz kelompok) diperoleh menggunakan carian manual grid. Untuk mengesahkan model yang dicadangkan, empat set data penanda aras yang terkenal dan satu set langkah penilaian digunakan. Kaedah yang dicadangkan kemudiannya dibandingkan dengan model sedia ada dan keputusan pada setiap set data telah direkodkan dan menunjukkan keberkesanan model yang dicadangkan. Model yang dicadangkan berjaya mencapai peningkatan dalam skor ketepatan ujian masing-masing 1%, 2%, 15% dan 5% pada digit MNIST, Fashion MNIST, X-ray Dada Pneumonia dan Set Data Pengesanan Emosi Muka.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACGAN	Auxiliary Classifier GAN
AI	Artificial Intelligence
BigGAN	Big Generative Adversarial Network
CapsNet	Capsule Neural Network
cGANS	Conditional GANs
CNN	Convolutional Neural Networks
DCGANs	Deep Convolutional Generative Adversarial Networks
DenseNet121	Densely Connected CNN
DL	Deep Learning
ECGAN	External Classifier GANs
EMD	Earth-Mover Distance
GANS	Generative Adversarial Networks
GCN	Graphic Convolutional Networks
GD	Gradient Descent
GS	Grid Search
GWO	Grey Wolf Optimizer
ML	Machine Learning
MLP	Multilayer Perceptrons
PBT	Population-Based Training
PCA	Principal Component Analysis
SCA	Sine Cosine Algorithm
SMOTE	Synthetic Minority Oversampling Technique
StyleGAN	Style Generative Adversarial Network
VG	Vanishing Gradient
VGG-16	Convolutional Neural Networks
WGANs	Wasserstein Generative Adversarial Networks

LIST OF SYMBOLS

f	Arbitrary function;
E	Electric Field
E_0	Electric field magnitude in V/m
e	Euler's constant
F^{-1}	Fast Fourier transform
F	Frequency
B	Linear Operator
C	Set of Complex numbers
c	Speed of light $(3\times10^8 \text{ m/sec})$

CHAPTER 1

INTRODUCTION

1.1 Introduction

Since the invention of Generative Adversarial Networks (GANs), they have been extensively and almost exclusively applied to image generation. GANs are trained using two adversarial networks, a Discriminator, and a Generator, working against each other under a min-max objective. Since 2014 many different variants of the GAN (Goodfellow et al., 2014) model have been developed to improve the image generation task, e.g., StyleGAN (Karras et al., 2018). StyleGAN is the architecture behind the famous website (Vincent, 2019), which can generate realistic human faces that do not exist. Another favorite variant of GAN is Wasserstein GAN(WGAN) (Arjovsky et al., 2017).

In the original GANs or Deep Conventional Generative Adversarial Networks (DCGANs), the Jensen-Shannon divergence is minimized to fool the discriminator; hence, it cannot distinguish between real or fake images, but in the case of WGANs, the Earth-Mover Distance (EMD) is minimized instead. This small change results in much better and more stable image generation result. The potential improvement on GANs is the Big Generative Adversarial Network (BigGAN) (Brock et al., 2018) which is the most recent cutting-edge model applied on the ImageNet model (Deng et al., 2009). The other model is a Progressive GAN, where the authors add new blocks of convolutional layers to both the generator and the discriminator models, which then take help from the actual image samples and try to generate images of high resolution from them and Pix2Pix GAN (Isola et al., 2017) which has several exciting applications such as edge-maps to photo-realistic images (Zhao et al., 2019). Also, advanced technological tools such as Web 3.0 manipulate images to increase their quality or extract useful information from them. However, to optimize efficiency and avoid time waste, images must be processed post-capture at a post-processing step (Albaom et al., 2021). Previous studies (Sun, Dai, Zhang, He, et al., 2021) indicated that “due to the small size and dense distribution of objects in UAV vision, most of the existing algorithms are effectively difficult to detect while GAN can be used to generate more synthetic samples, including samples from different perspectives and those from the identical perspective yet having subtle distinctions, thus enabling deep neural network training.”

1.2 Research Background

Semi-supervised learning is a machine learning technique where a small portion of data fed to the model is labeled during training, and the rest of the data is unlabeled. The two types of learning are semi-supervised learning (with a small amount of labeled training data) and supervised learning (with only labeled training data). It's a unique case of insufficient supervision. Semi-supervised learning is considered the way to go in many recent problems. It curbs many issues related to overfitting training data due to vast amounts of unlabeled noisy data. There is a lack of labeled data many times is another reason why Semi-Supervised Learning is gaining popularity these days since this requires human annotators, specialized equipment, and time-consuming tests (Zhu & Goldberg, 2009).

All Generative Models are applications of Semi-Supervised Learning. Here, the labeled samples are the real samples, while the generated samples are unsupervised. Selecting the accurate hyperparameters for deep learning or machine learning models is the important technique to excerption of the final squeezer out of the models. Searching manually for the right combination of hyperparameters is a very hectic task. Hence the author used various mathematical computation methods and algorithms for performing Hyperparameter Tuning. Some computational methods include Random Search (Bergstra & Bengio, 2012), Grid Search (Shekar & Dagnew, 2019). Then there are some algorithms like Population-Based Training (PBT) or Bayesian Optimization and Hyperband Method (BOHB), which have been utilized for optimization. Here in this thesis, the proposed simple Semi-Supervised Generative Adversarial Network Classifier (SGAN) has been trained by four multimedia benchmark datasets:

1. MNIST Digits Dataset (Deng, 2012).
2. MNIST Fashion Dataset (Xiao, Rasul, & Vollgraf, 2017).
3. Pneumonia Detection from Chest X-Ray Images Dataset (Mooney, 2018).
4. Facial Emotion Detection Dataset (Sambare, 2021).

Therefore, once a working baseline was set up, the researcher of this study progressed towards finding an optimal set of hyperparameters, like the learning rate for both the generator and the discriminator models. The need to optimize simple SGAN model would contribute to overcoming the problem of highly sensitive to the hyper parameter selections. Therefore, the researcher further applied a hybrid metaheuristic optimization algorithm called Sine Cosine Algorithm (SCA) for the hyperparameter tuning to obtain better results on the same datasets more efficiently as compared to using Grid Search and Manual optimization. Finally, the researcher has been compared the final proposed model with different deep learning methods that have been used to perform classification in the literature.

Sine Cosine Optimization Algorithm (SCA), a novel population-based metaheuristic algorithm which employs simple sine and cosine mathematical functions to find the global optimal solution (Mirjalili, 2016). The researcher used sine cosine mathematical equations to fluctuate towards or outwards in order to find the optimal solutions and utilized random variables to adaptively ensure this technique emphasizes the exploitation and exploration to find the possible global optima on the search space (Mirjalili, 2016). SCA has been found more efficient than other population-based algorithms in achieving an optimal global solution (Somu & Ramamritham, 2020). Different millstones on the search space are investigated when the sine and cosine functions return values greater than one or less than one. Thus, the researcher has chosen SCA due to its special characteristic which could be summarized as follows:

- The potential to escape the local optima, besides the high exploration inheritance of sine and cosine functions.
- The basic sine cosine functions enable this algorithm to adaptively shift from exploration $[< 1, > 1]$ to exploitation $[-1, 1]$.
- The inclination towards the finest region of the search space as the solution modifies its location around the finest solutions gained so far.

1.3 Motivation

There are two main avenues for improvement in a GAN-based class, improve the GAN as a whole and/or target the specific class labels and generate samples accordingly. Conditional GANs (Mirza et al. 2014) have been proposed for the same. Additional information, such as class labels connected with the input images, can be used to improve the GAN. This enhancement could be more ongoing training, faster training, or higher-quality generated images. From this concept of Conditional GAN Architectures (cGANs), researchers have been able to develop methods for semi-supervised generative classification of generated images like SGAN (Chavdarova & Fleuret, 2018), Auxiliary Classifier GAN (ACGAN) (Odena et al., 2017), and External Classifier GANs(ECGAN) (Haque, 2020). The discriminator of a typical GAN discriminates between real and fake samples. Then the same architecture can be used to build a classifier with just the output layer different via transfer learning. The researcher can classify the generated images into classes as required during the semi-supervised training of the SGAN. The discriminator model is updated in the Semi-Supervised GAN to predict the labels for the classes needed as the output alongside the prediction for the Real/Fake labels. Hence, the researcher is enabled to train the same discriminator model and the classifier model to predict the classes as well. That is how the semi-supervised GAN classifiers work. Similarly, there is a constant urge for improvement in the field of image classification, which led to the development of two improved types of GAN classifiers like Auxiliary Classifier GAN, which has two separated dense output layers. The first layer is Sigmoid that is using for identifying the fake/real samples and the second layer is Softmax which is using for the multiclass classification task. In July 2020, High School Student Ayaan Haque developed the External Classifier GAN, which contains three models - Generator, Discriminator, and an external multi-class classifier (Haque, 2020).

1.4 Problem Statement

Training of GANs is pretty hard at times which might be due to various problems and unfavorable conditions like:

Highly sensitive to the hyper parameter selections

The Google Brain paper indicates GAN is sensitive to hyperparameter tuning. The benefit of hyper parameter optimization can overtake the benefit of cost function if it is not optimized properly. Many cost functions demonstrate a wide range of performance with different hyperparameter settings. This can be desperate when the developers don’t know whether the models are not working or they need to engage in lengthy tuning (Lucic, Kurach, Michalski, Gelly, & Bousquet, 2018).

The figure 1.1 proves that the GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than others. The Black stars indicate the performance of suggested hyperparameter settings (learning rate & batch size) that applied for both the models’ generator and discriminator and then validated with different datasets like MNIST and FASHION MNIST (Lucic, Kurach, Michalski, Gelly, & Bousquet, 2018).

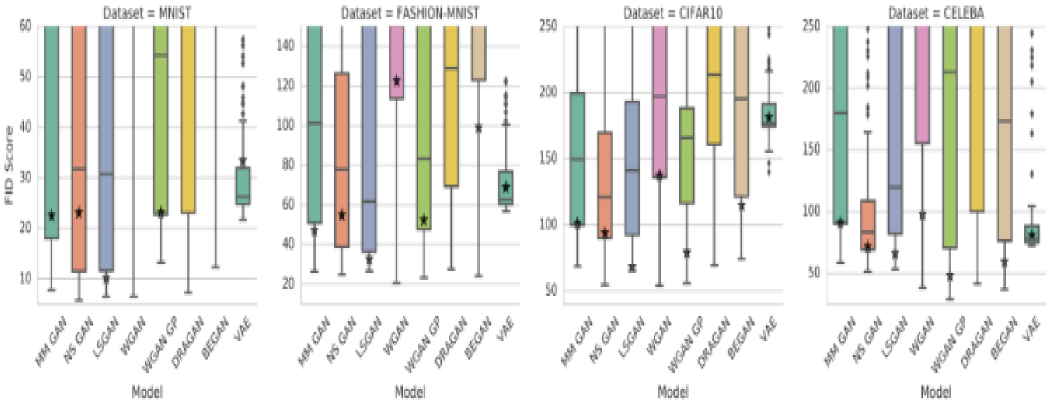

Figure 1. 1: Wide range hyper parameter search samples per model(Lucic, Kurach, Michalski, Gelly, & Bousquet, 2018).

Another example in the figure 1.2 indicates a (y-axis), that is the performance between an (x-axis) which is the different learning rates under various cost functions (LSGAN, WGAN, WGAN-GP, BEGAN etc) that may cloud the model judgment in whether the model design is working (Hui, 2018).

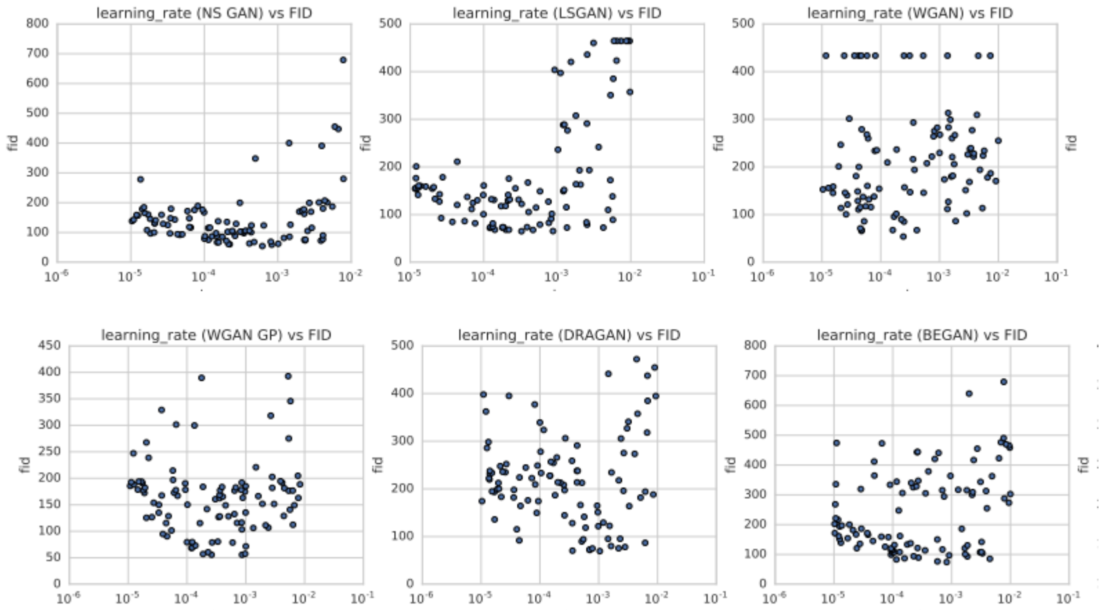

Figure 1. 2: GAN Learning Rates (Hui, 2018).

During the training of the GAN, the hyperparameter tuning requires patience, so no cost functions are going to implement without taking time on the hyperparameter tuning since many new cost functions will be introduced hyperparameters with sensitive performance (Hui, 2018).Vanishing gradient exists during the training process of GANs due to the multiplication of the gradient with small values through the backpropagation process. This causes the GAN Network to stop learning and gives bad accuracy (Ribeiro et al., 2020). Therefore, enhancing

GANs networks' architectures will lead to better performance in terms of prediction accuracy. There are many essential influences in GANs, which can be explored for a particular task to improve the performance of prediction accuracy, where generating the GANs weights based on meta-heuristic algorithms like the Sine Cosine Algorithm (SCA) instead of traditional manners will lead to better performance as well as overcoming the limitations of existing work (Somu & Ramamritham, 2020). This is the overview of what the researcher is trying to achieve through the weight initialization of the discriminator and the classifier networks of the GAN.

1.5 Research Questions

Based on the problem and motivation mentioned above, this research is going to concentrate on finding the appropriate answers to the questions listed below:

1. How to determine the SGAN hyperparameter selection sensitivity issue during the model training process.
2. How to propose an algorithm for optimal selections of hyperparameters during training the process of SGAN?
3. How to develop an optimized SGAN model that solves the problem of sensitivity to hyperparameter selections?
4. How to validate and test the proposed model using four benchmarking multimedia datasets and existing evaluation classification metrics?

1.6 Research Aim

This study aims to find out the core architectures of the Semi-supervised GAN Classifiers over four different multimedia benchmarking datasets (MNIST Digits, MNIST Fashion, Pneumonia Detection from Chest X-Ray images, Facial Emotion Detection) and enhance these architectures with a metaheuristic optimization algorithm named Sine Cosine algorithm (SCA) to be used for weights and biases initialization.

1.7 Research Objectives

The researcher has identified the below research objectives to achieve the research aim stated in Section 1.6:

1. To determine the SGAN hyperparameter selection sensitivity issue during the model training process.

2. To develop and enhance the SGAN model based on the SCA algorithm for the optimal selections of hyperparameters during the training process of SGAN.
3. To validate and test the optimized SGAN model for image classification using four multimedia benchmark datasets and existing evaluation classification metrics.

1.8 Research Scope

This research falls under the domain of Deep learning that investigated the previous attempts and their model's efficiency in predicting the SGAN based only on the problem of hyperparameter selections. It also focused on developing and implementing the Sine Cosine algorithm (SCA) for the optimal selection of hyperparameters during the training process of SGAN. The proposed SGAN model used for classification problems and validated using benchmark multimedia image datasets only and tested using existing evaluation classification metrics for model validation. The research will use existing evaluation classification metrics for model validation. Overall, the research scope will be:(i) Deep learning, (ii) Affection of hyperparameters in Semi-Supervised Generative Adversarial Networks, (iii) Public multimedia image datasets for classification problems and existing evaluation classification metrics for model validation. Figure 1.3 shows the main scope of this research.

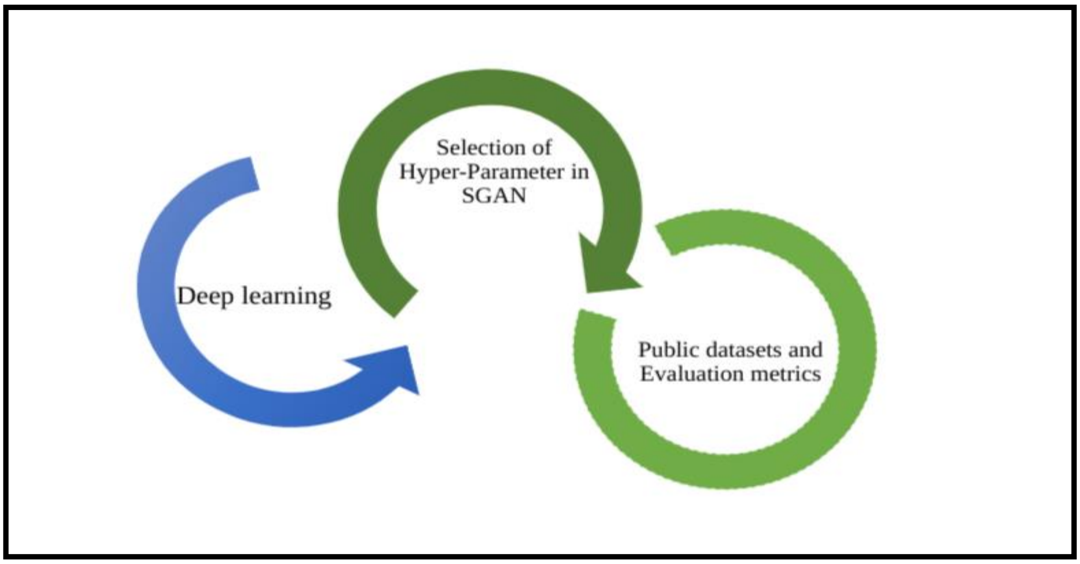

Figure 1. 3: The scope of the research

1.9 Significance of Study

This research contributes to identifying the effects of trainable parameters initialization in Semi-supervised generative adversarial networks by introducing the Sine-Cosine algorithm

(SCA) for optimal weights selection and biases during the training process SGAN. The proposed algorithm is expected to be applied for classification problems using image datasets only and applied for existing evaluation classification metrics for model validation. This research can be act as a fundamental and director for future researches of semi-supervised generative classification of generated images based on the use of Sine-Cosine algorithm (SCA).

1.10 Thesis Contribution

The main contribution of this study is to enhance the hyperparameter of semi-supervised GAN classifiers-based Sine Cosine Algorithm for multimedia datasets." Here the architecture is used for binary and multiclass image classification on four benchmark datasets. The contributions that are obtained in this thesis are:

1. Extensive, exhaustive literature and code reviews have been done to use different weight optimization techniques for the SGAN classifiers identify the research gaps that exist in the literature.
2. Enhanced the weight-optimized SGAN classifier based on Sine-Cosine Algorithm optimization for Multimedia datasets.
3. The Grid Search (GS) was efficiently applied for each experimental setup model to find the best optimal set of hyperparameters (Learning rate & Batch size) by locating and picking the most appropriate values for the dataset size and model structure where the researcher gets the most optimal results.
4. Synthetic Minority Oversampling Technique (SMOTE) used to address class imbalances in the unbalanced datasets which further led to a substantial improvement in the proposed model's performance.
5. The final proposed model enhanced by the Sine Cosine Algorithm (SCA) for the optimal selection of hyperparameters during the training process of the final model which represents the novelty and originality of the study.
6. Four well-known multimedia benchmark datasets (MNIST digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection) and a set of existing evaluation classification metrics were used to validate and show the efficiency of the proposed model.
7. The proposed model successfully showed improvement in the test accuracy scores on benchmarking multimedia datasets respectively.
8. The proposed model was compared against existing models, the results on each dataset were recorded and demonstrated the effectiveness of the proposed model in

the classification accuracy in many cases.

1.11 Thesis Structure

The thesis consists of 5 main chapters including the Abstract at the start and the References as well as the Appendix containing the code base behind this thesis.

Chapter 1 describes the study's baselines. It is organized into ten sections, with sub-sections in some of them. The first section, 1.2, discusses the backdrop of the research. The motivation for the project is discussed in section 1.3. The section 1.4 talks about the research problem statement. Sections 1.5 and 1.6 discuss the research questions and goal, respectively, while the research objectives were stated in section 1.7. The remaining sections demonstrate the research's scope 1.8, significance 1.9, and key contribution (1.11). The thesis' structure is summarized in section 1.11.

Chapter 2: Literature review discusses the study literature relevant to that conducted in this thesis. There are several sections in this chapter. The processing of time-series data is discussed in section 2.1. Section 2.2 provides an overview of artificial intelligence. An overview of machine learning techniques is presented in section 2.3. Deep learning is discussed in section 2.4. The recent relevant work is summarized in section 2.5. The methods for weight optimization are discussed in section 2.6.

The approach of the suggested model, which is separated into phases, is explained in Chapter 3. It explains how the investigation was carried out. In addition, the evaluation metrics used to assess the proposed alternatives are detailed in detail. This chapter also includes the benchmarking datasets that were utilized for validation. There are several sections in this chapter. The research approach is examined in Section 3.1. Section 3.2 lays out the methodology flowchart used to meet the research goals and a full description of each stage in the flowchart. The environment used to implement the proposed methods is depicted in section 3.3. The datasets and the models used in this study were reported in section 3.4. Section 3.5 explained the process of preparing the data. The chapter then ends with a summary in section 3.6.

Chapter 4 reports the outcomes of this study from the proposed methodology, and also a comparison to existing methods is discussed. First, it presents how the experiments have been conducted and illustrates the results of each experiment model. It also shows the results that obtained by the proposed model (SGAN+SMOTE+SCA). A comparison of the proposed model results with the previous models has been conducted and discussed in this chapter via section 4.5, followed by the significance analysis 4.6.

Chapter 5 concludes the thesis with a discussion of the thesis work and the future works that might be possible in this field of study.

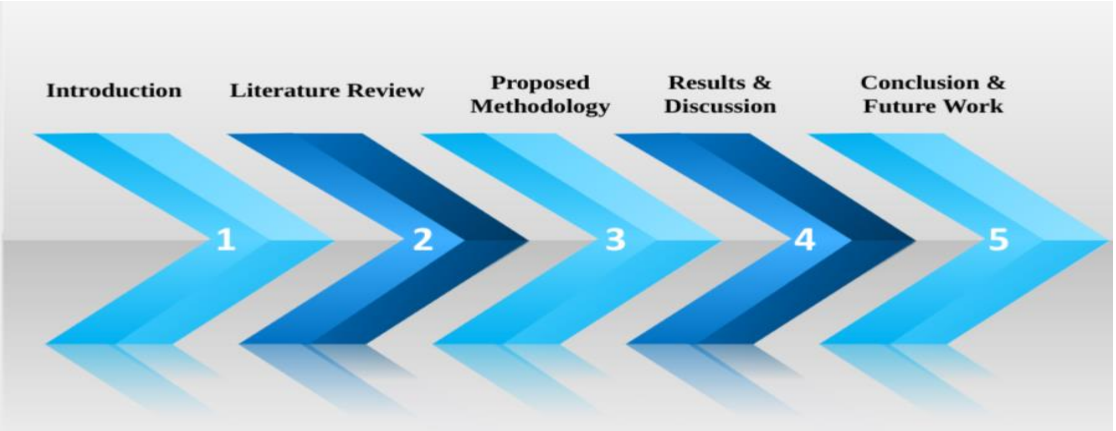

Figure 1. 4: Thesis Structure

CHAPTER 2

LITERATURE REVIEW

In this chapter, the researcher reviews the literature on which this thesis work is based. All the necessary background research and theory are mentioned in this section of the Literature Review. The researcher starts from basic concepts like Machine Learning and Artificial Intelligence to Semi-Supervised GANs and Sine-Cosine algorithms.

2.1 Introduction

For the last 50 to 60 years, the researchers have witnessed the growth of the Data Revolution. With the advent of Machine Learning, starting with the famous question of “Can Machines learn?” by Alan Turing, Machine Learning has been on the upward journey. It all started with a paper that believed that machines could distinguish between human and machine-generated texts when a human finds difficulty (Petzold, 2008; Machinery, 1950).

Now, in the 21st century, machine learning is the way to go for many tasks varying from common problems like regression, classification, and clustering to applications in fields of science, military, space, geography, geology, medicine, surgery, etc. and social media among others. Today, machine learning has found its place as an integrated service in so many places that it’s almost impossible to find places where ML is not used and where researchers do not leave data footprints. Machine-learning algorithms are used to identify objects in photos, convert speech to text, match user-interested news items, posts, or products, and choose appropriate search results (Turing, 2009; Zhang, Lipton, Li, & Smola, 2021).

Although Machine Learning is a new field, its natural history dates back to the 20th century. Multilayer Perceptron (McCulloch & Pitts, 1943), Q-Learning (Watkins & Dayan, 1992), long short-term memory (Hochreiter & Schmidhuber, 1997), and convolutional neural networks (Maier & Dandy, 1998) are some topics that had been proposed way back in the 20th century but are in very heavy use in modern research and industry. Sequential modelling and prediction of time-series or text data, which appear as streams or chunks of unstructured data, have been an important topic. The feed-forward networks and algorithms (Dietterich, 2002; Bengio, Vinyals, Jaitly, & Shazeer, 2015; Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019) were unable to capture the history or sequence information. Hence, they could not capture information in processing the sequential data in text analysis, image captioning, signal data, and time-series data (Längkvist, Karlsson, & Loutfi, 2014; Alom et al., 2018).

With the emergence of recurrent neural network algorithms (Rumelhart, Smolensky, McClelland, & Hinton, 1986; Yu, Li, Bao, Tang, & Zhai, 2018), sequential modeling has become possible since they have a hidden memory called “cells” that stores temporal dependencies from previous time-steps and link them with predicted output (Petneházi, 2019). However, the RNNs suffer from the problem of vanishing gradients, due to which they cannot hold some information for a long time. This problem was addressed and dealt with in (Hochreiter & Schmidhuber, 1997; Li, Li, Cook, Zhu, & Gao, 2018), where the authors proposed the idea of Long Short-Term Memory which was capable of storing the history along with the short-term memory of the RNNs. (Kumar, 2017) dealt with how and when the information in the cell state of the LSTM should be flushed out. LSTM, on the other hand, has a high cost of computation because of its large number of gates (Salehinejad, Sankar, Barfett, Colak, & Valaee, 2017; Cho et al., 2014; Werbos, 1990). The researcher gradually moves into an even younger subfield of Machine Learning known as Deep Learning, on which this thesis is based (LeCun, Bengio, & Hinton, 2015).

2.2 Artificial Intelligence (AI)

Planning, reasoning, solving problems, comprehending complicated ideas, thinking abstractly, learning quickly, and learning from experience are all constituents of AI (Holmgren & Morgan et al., 1976). The entire map of Artificial Intelligence is shown below in the figure. Machine Learning is the subfield of AI that deals with applications of specific task-oriented mathematical algorithms to learn different underlying patterns in the given data. It improves performance by training the machine learning model on the provided data over time. Machine Learning can generally be classified into three broad types: Supervised, Semi-Supervised, and Unsupervised Learning. Deep Learning is a subset of machine learning that deals with the same problem of finding hidden patterns in the data but with neural networks to get better results over the generic machine learning algorithms used on the same issues. DL is explained in detail in Section 2.4.

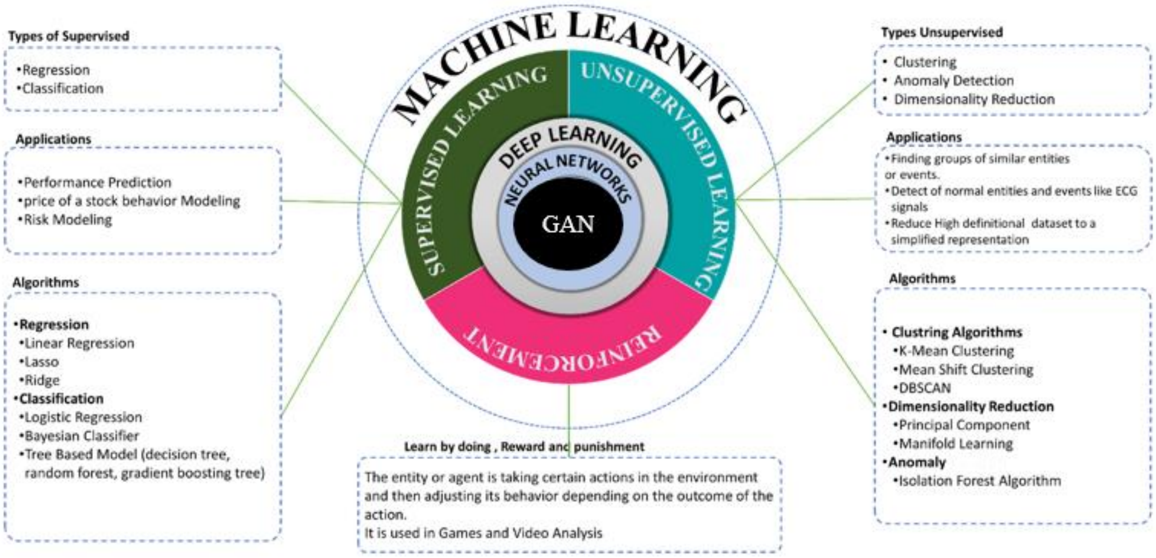

Figure 2.1: Types of Machine Learning and Artificial Intelligence

2.3 Machine Learning

As mentioned before, supervised, semi-supervised and unsupervised learning are three categories of machine learning. However, the researchers can also have a mix of 2 types of learning. For example, the most famous type is semi-supervised learning (between supervised and unsupervised learning) (Abdi, Shamsuddin, Hasan, & Piran, 2018).

2.3.1 Supervised Learning

In this type of ML, the dataset provided for training contains both the data features and the correct output labels based on which the model is trained to utilize a suitable algorithm and loss function, and then the outputs for some test data are predicted, which was not seen before (Rashidi, et al., 2021).

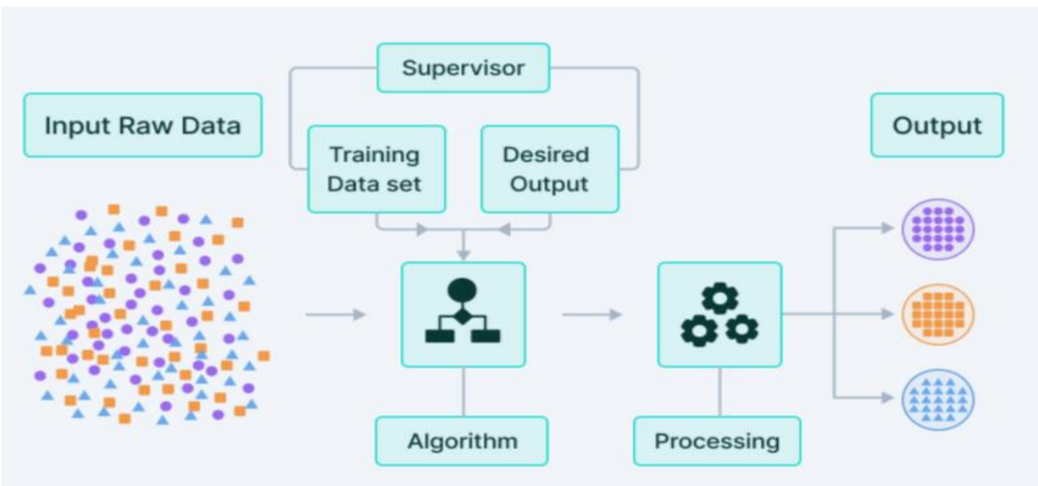

Figure 2. 2: Supervised learning (Rashidi, et al., 2021).

As shown in Figure 2.2, evaluating and constructing a mapping function via supervised learning is applied via utilizing two stages. The first stage is dividing the dataset into two groups of samples, the training dataset and the testing dataset. Through this stage, the test vector (the inputs) with one or more perfect output values are identified in both training and the test results. The mapping feature is trained with training data collection until it gains substantial performance (a metric for how precisely the mapping function maps the training data to the associated demanded output). This occurs for all training samples in the shape of supervised learning were utilizing the function cost to allocate the error (actual vs. expected output) to modify the mapping function. After that, the test dataset is tested against the learned mapping feature. This test dataset includes an excellent indicator of how well the mapping algorithm generalizes to new data and data not used during testing.

Typical tasks of this type of learning are classification and regression (Hadi & Chatterjee, 2015). Figure 2.3 illustrates the contrast between classification and regression learning.

1. **Regression:** A regression problem is when the output variable is a natural or continuous value. There are many different types of regression linear, logarithmic, polynomial, etc. (Chatterjee & Hadi, 2013). The best possible fit is when the line goes through the data points in the hyper-plan (Moisen, 2008).
2. **Classification:** When the output variable is categorical, the problem is called a classification problem. A classification model tries to deduce something from seen data. A classification model will attempt to predict the value of one or more outputs given one or more inputs (Abdi, Shamsuddin, Hasan, & Piran, 2018).

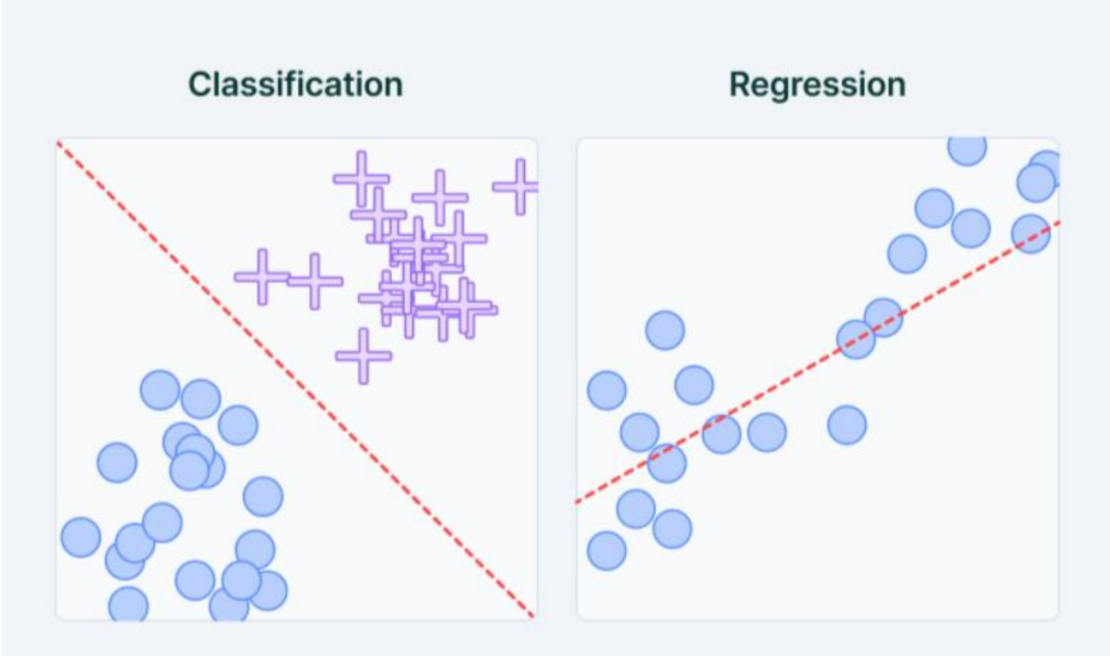

Figure 2. 3: Regression vs. Classification in Machine Learning (Bozkurt, 2022).

2.3.2 Unsupervised Learning

In unsupervised machine learning, clusters of unlabeled datasets and analyses utilize machine learning techniques. Without human intervention, these algorithms help data grouping or uncover patterns. Clustering (KNN, K-Means Clustering), dimensionality reduction (PCA, SVD), and association (mainly Apriori algorithms) are using unsupervised learning models (Abdi, Shamsuddin, Hasan, & Piran, 2018; Moisen, 2008; Barlow, 1989).

- **Clustering:** This method is a data-mining task that arranges the unlabeled data into a set of collections rely on differences and similarities. The clustering methods are utilized to arrange the unclassified data items and raw them into series, identified via patterns or information structures. Overlapping probabilistic, including exclusive and hierarchical, are the core kinds of the clustering algorithm (Barlow, 1989).

.

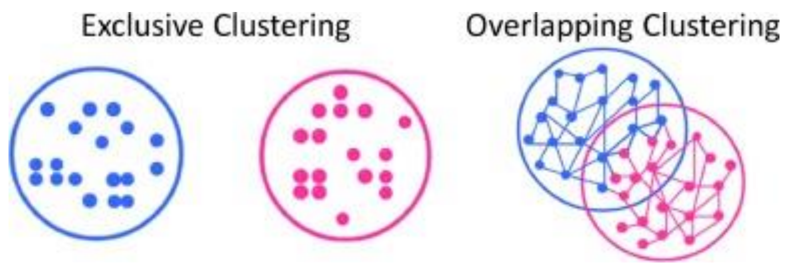

Figure 2. 4: Diagram representing Overlapping Clustering (Khanmohammadi, Adibeig, & Shanehbandy, 2017).

- **Association** is a rule-based technique of detecting associations between variables in a dataset is an association rule (Barlow, 1989). Market basket analysis lets organizations better understand connections between typically employed methodologies and different items. Businesses may build huge solid cross-selling tactics and recommendation engines via grasping their customers' consumption patterns; an example of this technique is the playlist of Amazon's "Customers Who Bought This Item Also Bought" and Spotify's "Discover Weekly." Many algorithms extensively utilize the Apriori technique for producing association rules, including FP-Growth, Apriori, and Eclat

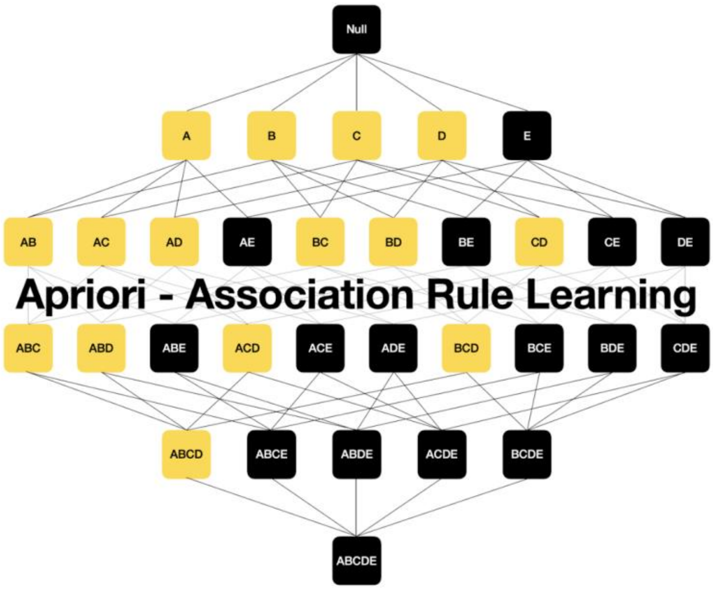

Figure 2.5: Diagram representing Apriori Association(Borgelt, 2003).

- **Dimensionality reduction:** Through this method**,** more data generally gives huge accurate findings, but it can also affect the effectiveness of machine learning algorithms like overfitting and creating visualization datasets most challenging. The dimensionality reduction method is performed when dimensions and number of characteristics in a dataset are too huge because it increases the data quantity inputs to a sensible standard. In contrast, the dataset's integrity is maintained to the largest scope feasible. It's usually employed at preprocessing step of the data. Apart from that, PCA, SVD, and Auto Encoders are the famous types of dimensionality reduction algorithms (Abdi, Shamsuddin, Hasan, & Piran, 2018; Moisen, 2008; Barlow, 1989).

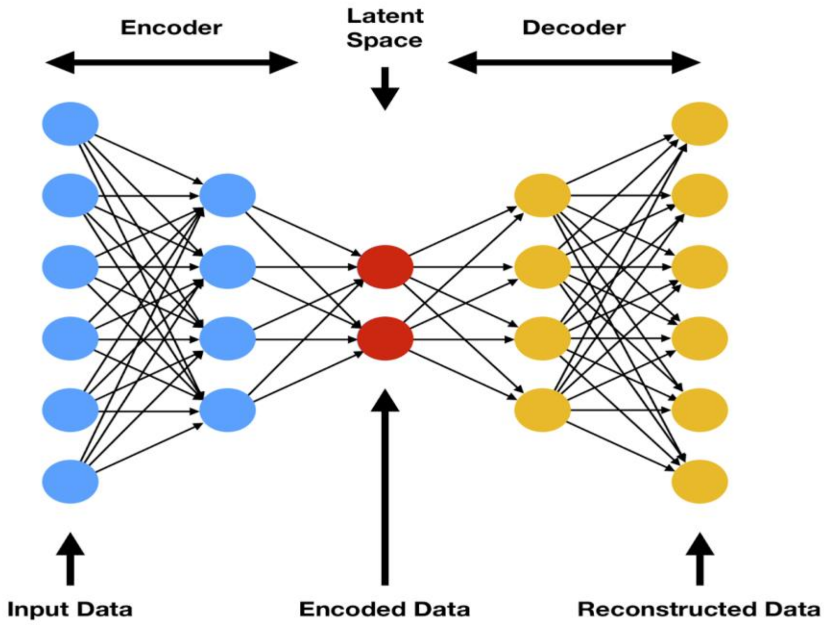

Figure 2.6: Diagram representing Variational Auto Encoders (Flores, 2019).

2.3.3 Reinforcement Learning

Reinforcement learning (RL) is a type of machine learning that focuses on how intelligent agents must pick actions in the environment to increase the accumulative reward and reduce the penalty. Reinforcement learning is considered a natural learning process like it's the way humans or any other intelligent agent tends to learn. (Abdi, Shamsuddin, Hasan, & Piran, 2018). Reinforcement learning is a type of machine learning model training used to create groups of judgments. Aside from that, the agent grasps to earn a goal in an uncertain and possibly complicated. In reinforcement learning, intelligent agent meets up a game-like circumstance. Therefore, the computer utilizes error and trial to discover the problem solution. Either penalties or incentives are introduced via intelligent agent for the act. It gathers to identify the accomplishment of what the programmer desires and aims to maximize the final prize as much as possible.

Even though the designer defines the reward policy, the rules of the game supply the model with no idea or tips for how to complete the game. Progressing and random trials are starting to compound superhuman abilities and strategies, and this step relies on the model to discover how to perform the task to increase the reward. Reinforcement learning is presently considered the ultimate effective method to trace the machine's creativity via utilizing numerous trials and search power. Unlike humans, Intelligence agent might earn experience provided from huge simultaneous gameplays. A reinforcement learning algorithm is implemented on solid computer infrastructure.

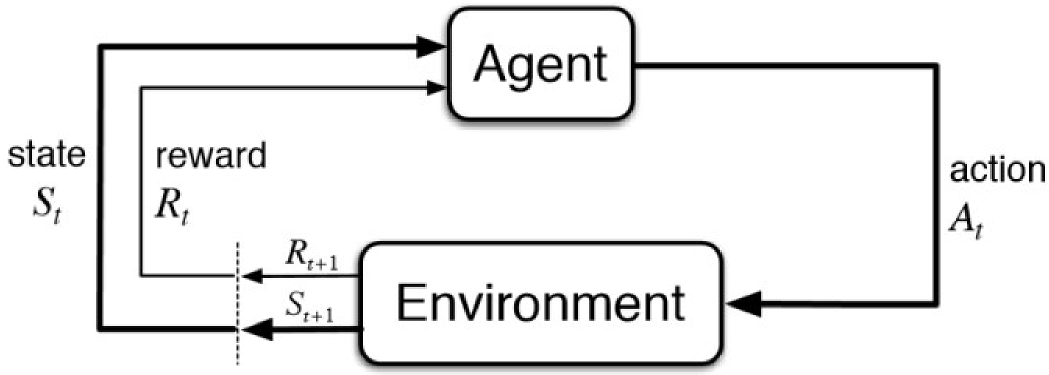

Figure 2.7: Reinforcement Learning Block Diagram (Abdi, Shamsuddin, & Piran, 2018).

2.4 Deep Learning

Deep Learning is a subpart of machine learning that bargains with the problem of predicting and discovering hidden patterns in the data but with a different approach. Here, Neural Networks are used to simulate the functioning of the animal brain and solve a particular data-driven problem. There are many vital algorithms in deep learning like the Multi-Layer Perceptrons, ANNs, CNNs, RNNs, and their variants. Also, there are Generative Adversarial Networks which is the main focus of this thesis to rely on, and these are the fundamental algorithms of deep learning. There are advanced variants of these algorithms, and there is space for constant growth in this field.

2.4.1 GANS and Related Works

In the generative adversarial networks (Goodfellow et al., 2014), there are two models, which are the generator(G) and the discriminator(D). These models are put up against one another in an adversarial fashion. The task of the G model is to take input of random noise as input and output a fake image which is then fed to the D model, and the D model tries to classify whether the image it got from the generator is real or fake. The task of the G model is to be able to fool the D model and to be able to generate realistic fake images after epochs of training and backpropagation. The GAN architecture and the entire classification process are shown in figure 2.8.

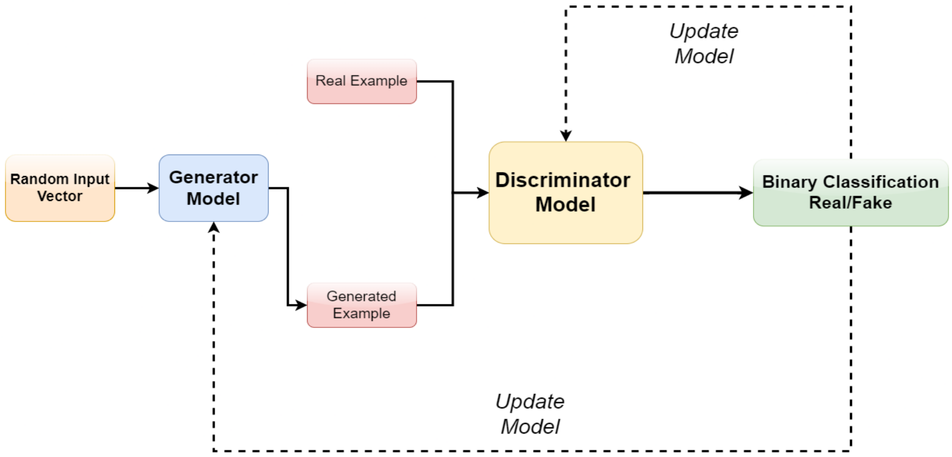

Figure 2.8: Generative Adversarial Network Architecture (Goodfellow et al., 2014).

GANs were generally used for image synthesis from random noise hence were applied to different fields like fake image synthesis, data augmentation, conditional image synthesis. But recent advancements in GANs have resulted in the diversified application of GANs. One such application is the classification of fake images generated by the Generator model. This field exploded with the advent of Conditional GANs. Conditional GANs are an extension of the min-max Generative modelling (GANs) whereby the developers pass the required class as a part of the input to the Generator and the Discriminator, respectively, along with the random noise and the fake sample generated respectively. This results in the generation of images belonging to a particular class as desired (Mirza & Osindero, 2014). Though preliminary, this thesis opens up a whole new domain of interesting application of GANs.

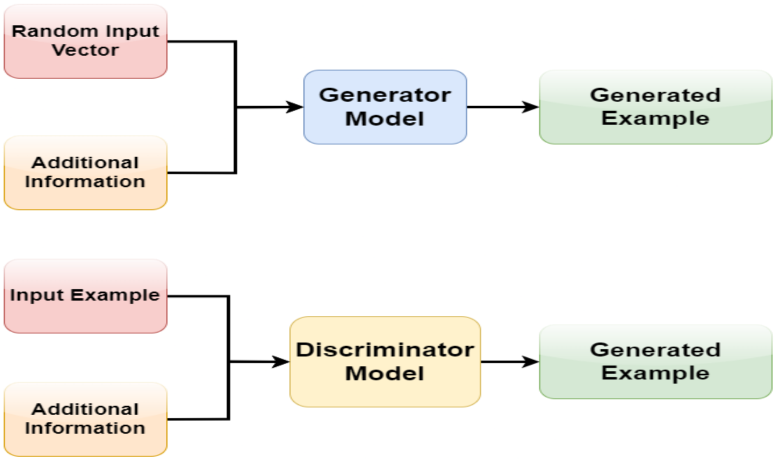

Figure 2.9: Conditional GAN Architecture(Odena, Olah, & Shlens, 2017).

After this, many new GAN architectures were designed pertaining to the problem of image classification (both supervised and semi-supervised). Some of them are discussed below:

- **Semi-Supervised Classification with GANs(SGAN)**: The researchers attempt to achieve the task of classification of a data point x into K classes using a standard classifier as an extension of the discriminator model. Thus, the discriminator model is designed to contain multiple output channels, i.e., (1) Binary Classifier for identifying fake vs. real samples (2) K channels consisting of probabilities derived from applying softmax activation function to the elements of the K-dimensional output vector. These K channels represent the probability of the generated sample belonging to each K class. This technique is widely known as SGAN (Salimans et al., 2016; Odena, 2016). However, scalability is one of challenging issue that has been founded in SGAN (Reddy, Viswanath, & Reddy, 2018). Nowadays, SGAN has a wide range of application scenarios and has attracted much attention. However, it is noteworthy that although the learning performance is expected to be improved by exploiting unlabeled data. Some empirical studies show that there are situations where the use of unlabeled data may degenerate the performance (Li & Liang, 2019).
- **Auxiliary Classifier GANs(AC-GAN):** The AC-GAN is another type of GAN classifier that changes the discriminator such that instead of taking the class labels as input (which was the case in Conditional GANs) and predicts the class labels of the generated image that was passed into the discriminator with

Auxiliary Classifier. It makes the training process stable, and hence the generator can generate high-quality images when its weights and biases get trained through forward and backward propagation. In this method, the authors pass the class labels along with the random noise at the Generator end, but unlike the cGANs, the authors do not pass the labels at the Discriminator end. As before, the discriminator model must estimate whether the input image is real or fake and the image's class label. There are two dense layers, one for the sample classification into fake or real and the other for categorical multi-class classification into K classes, to which the image belongs (Odena, Olah, & Shlens, 2017). However, the diversity of the generated samples by AC-GAN tend to decrease as the number of classes increases, hence limiting its power on large-scale data and this one outstanding limitation of AC-GAN approach. Another limitation is that the auxiliary classifier in AC-GAN imposes perfect separability which is disadvantageous when the supports of the class distributions have significant overlap (Gong, Xu, Li, Zhang, & Batmanghelich, 2019).

- **External Classifier GANs (ECGAN):** This is a semi-supervised GAN architecture where the fake image generated by the generator is used to improve image classification. Generally, the present models for classification with GANs (ACGAN, SGAN) have the same discriminator and classifier models, with the only difference being the output layer. EC-GAN attaches a GAN's generator to a classifier, hence the name, instead of sharing a single architecture for discrimination and classification. The promising results of the algorithm could prompt new related research on how to use artificial data for many different machine learning tasks and applications. Thus, there are three separate models - generator, discriminator and multi-class classifier. The Discriminator is trained classically for GANs and the same goes for the classifier. In ECGANs, all the real samples must have labels assigned to them. The generated images are then used as inputs for classification supplementation during training. This architecture follows a semi-supervised approach because the generated images do not have any labels. The generated images and labels are only kept if the model correctly predicts the sample class with a high probability (The labeling is done through a process of pseudo-labeling initially) (Haque, 2020). Including classifiers in cGANs often comes with a side effect of only generating easy-to-classify samples and this is a significant issue of ECAGN. Recently, some representative cGANs avoid the shortcoming and reach state-of-the-art performance without having classifiers. Somehow it remains unanswered whether the classifiers can be resurrected to design better cGANs (Chen, Li, & Lin, 2021).

Based on the limitations of the current GAN approaches (SGAN, ACGAN & ECGAN) which have been illustrated in the previous section, the researcher of this study has proposed an excellent simple SGAN architecture for the image classification task and performed weight initialization based on using Sine-Cosine Algorithm(SCA). Figure 2.10 is the proposed SGAN classifier architecture.

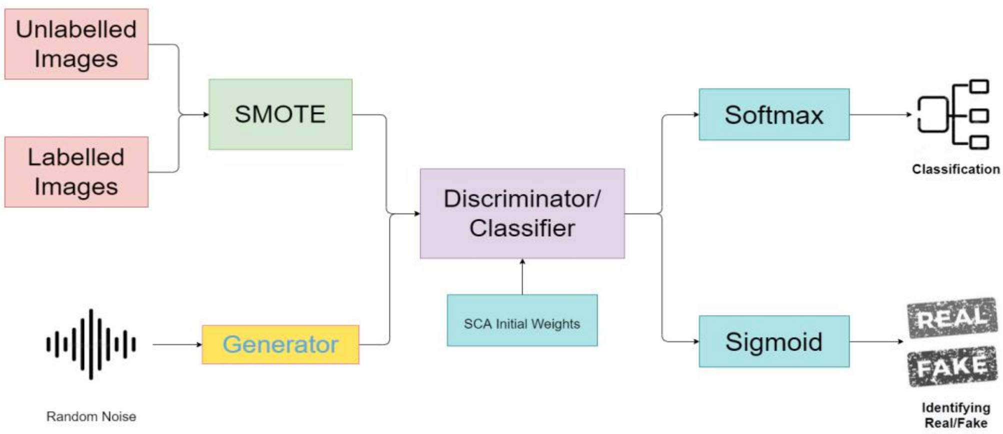

Figure 2.10: Proposed SGAN Classifier architecture

2.5 Optimization Algorithms

GANs are one of the most computationally intensive models to train, as it is the equivalent of training two neural networks at the same time. For the lousy portable computer, training the GAN until convergence is very difficult due to highly sensitive to hyperparameter tuning (Lucic, Kurach, Michalski, Gelly, & Bousquet, 2018). Hyperparameters are simply those parameters in a Machine Learning architecture that are not learned from the data during model training. Manually searching for the best hyper-parameter can be very tedious, and hence the researcher uses various algorithms to perform hyperparameter tuning. These include:

- **Grid Search:** The researcher iterates over all the possible combinations of the hyperparameter values in grid search. Then, the researcher runs the model training over each set of hyperparameters and then returns the set of hyperparameters for which the most optimal results have been achieved (Cherkassky & Ma, 2002).
- **Random Search:** This method is also like grid search, where the researcher produces

a possible set of hyperparameters values, trains the model on random sets of hyperparameters, and returns the set of hyperparameters that give the best results (Bergstra & Bengio, 2012).

- **Bayesian Optimization:** This algorithm tries to find the hyperparameters for which the loss is minimum in the minimum possible steps. (Brochu, Cora, & De Freitas, 2010; Frazier, 2018)

2.5.1 Gradient Descent

Gradient Descent is used to achieve the optimal trainable parameters by minimizing the cost function (Ruder, 2016). The feedforward and backpropagation algorithms train neural network models, and Gradient Descent is the most fundamental and widely used optimization during automatic differentiation for backpropagation (Ruder, 2016). However, due to vanishing gradient and gradient explosion, training GANs using GD is problematic (Chiroma et al., 2018).

Nowadays, there are many different Gradient-based optimizers, which are improvements over the original Gradient Descent algorithm (Bengio, 2012).

1. **Stochastic gradient descent:** This method is a computationally faster version of the original Gradient Descent Algorithm. It performs updates frequently since the updated space is each element of the data and not on the entire batch of data (Bottou, 2012).
2. **Mini batch:** Mini-batch GD is computationally cheaper than SGD and much faster than GD because the updates are performed on a subset of the batch at a time (Ruder, 2016).
3. **Momentum:** This method is a gradient descent optimization feature that allows gaining inertia in a special direction of the search space, i.e., towards the optimal solution, the minima, while resisting noisy gradient oscillations due to frequent updates (Martens & Sutskever, 2012).
4. **RMSProp:** In this improved version of Gradient Descent, the model is made to learn for an infinite time with improvements (Dozat, 2016).
5. **Adaptive Moment Estimation (Adam):** It records the moving average of the previous gradient values and tends to converge much faster than the standard RMSProp by applying Momentum to it (Kingma & Ba, 2014).
6. **AdaGrad:** This method is a parameter-specific learning rate optimizer that adjusts to the frequency with that a parameter is improved through the training. The smaller the

improvements become when a parameter receives more updates (Zeiler, 2012).

2.5.2 Metaheuristic Optimization

Weights optimization in GANs has not been studied extensively yet. However, a few methods, including initialization with Random Seed, genetic algorithms, or Meta-Heuristic Population-Based methods might be a potential solution to the trainable parameter's initialization problem in GANs. Metaheuristic optimizations play the traditional role of searching and finding the optimal solution for most of the complicated problems such as path planning, data clustering, feature selection (Al-Tashi, Rais, & Abdulkadir, 2018), image processing, and others (Mirjalili, Song Dong, Lewis, & Sadiq, 2020) (Mirjalili, Song Dong, Lewis, & Sadiq, 2020). These techniques can invent solutions or improve existing solutions. Metaheuristic optimization can be divided into two main types are Swarm based algorithms and Evolutionary based optimization (Nawi, Khan, Rehman, Chiroma, & Herawan, 2015). Aside of that, the researcher has summarized and analyzed the most existing trending metaheuristic optimization algorithms that may use to enhance the proposed SGAN model instead of using traditional manners. The types of metaheuristic optimization algorithms are:

1. **Particle Swarm Optimization (PSO)**: Particle swarm optimization (PSO) is inspired by the bird swarms algorithm, which simply looks for the core solution of the solution space. It differs from other optimization techniques; it has very few hyperparameters and simply requires the objective function, and is unaffected by the gradient or any differential form of the objective (Mirjalili, Song Dong, Lewis, & Sadiq, 2020).
2. **Ant Colony Optimization (ACO):** This method identifies the optimal solution in case of the most complex optimization problems (Dorigo, Birattari, & Stutzle, 2006; Al-Tashi, Rais, & Abdulkadir, 2018).
3. **Bat Algorithm (BAT):** It is used to find the solution to global optimization problems with considerable speed (Ma & Wang, 2018).
4. **Cuckoo Search (CS):** This method is using in the case of multimodal objective functions optimization and can work with considerably few parameters (Yang &

Deb, 2009).

5. **Grey Wolf Optimizer (GWO):** This method is also used in the case of finding the global optimum for the multimodal objective functions. However, this method suffers from multiple local optima in the case of unimodal objective functions (Mirjalili, Mirjalili, & Lewis, 2014; Al-Tashi et al., 2020).
6. **Sine Cosine Algorithm (SCA):** A metaheuristic population-based algorithm where many random agents search between upper-bound and lower-bound. The agents are tweaked to reach the required solution based on a mathematical construct sine and cosine functions (Mirjalili, 2016).
7. **Whale Optimization Algorithm (WOA)**: This optimization algorithm is useful in cases where the solution search space is unknown (Mirjalili and Lewis, 2016).
8. **Genetic algorithms (GA):** Genetic Algorithms are constrained minimization algorithms that work in the way natural evolution works. It chooses a few random agents and then tries to oscillate them towards the minima to find the optimal solution (Holland, 1992).

From all metaheuristic optimization algorithms types, the researcher found the Sine Cosine Algorithm (SCA) as an efficient and a practical search for weights initialization of the discriminator and the classifier models of the proposed model. SCA is a novel population-based metaheuristic algorithm which employs simple sine and cosine mathematical functions to find the global optimal solution (Mirjalili, 2016). The researcher used sine cosine mathematical equations to fluctuate towards or outwards in order to find the optimal solutions and utilize random variables to adaptively ensure this technique emphasizes the exploitation and exploration to find the possible global optima on the search space (Mirjalili, 2016). SCA has been found more efficient than other population-based algorithms in achieving an optimal global solution (Somu & Ramamritham, 2020). Different millstones on the search space investigated when the sine and cosine functions return values greater than one or less than one. Thus, the researcher has chosen SCA due to its special characteristic which could be summarized as follows:

- The potential to escape the local optima, besides the high exploration inheritance of sine and cosine functions.
- The basic sine cosine functions enable this algorithm to adaptively shift from exploration [$< 1, > 1$] to exploitation [-1, 1].
- The inclination towards the finest region of the search space as the solution modifies its location around the finest solutions gained so far.

2.6 Previous Related Works to Hyperparameters of GAN techniques

Generative Adversarial Network (GAN) is currently one of significant topics in Deep Learning as well as there are a massive increase in the number of papers which being published on GANs and its main problems over the last different months since GANs have been applied to a great several problems one of this problem is GANs highly sensitive to the hyperparameters selections due to GANs have a sophisticated design that can be challenging to train. This is because obtaining the proper set of parameters for all models-generator, discriminator, and classifier is complex. As a result, training a single GAN model for different datasets may not produce satisfactory results

Apart from that, the researcher conducted an extensive literature review on to the hyperparameters selections of GAN networks. Table 2.1 briefed the previous related works which have executed several techniques to enhance the hyperparameters of GAN. It summarized the matrices, benchmarking datasets, the result achieved and the limitations of these studies.

Table 2.1: Related works to hyperparameters of GAN techniques

Related Work, Author, Year	**Technique**	**Dataset**	**Metric**	**Results achieved**	**Limitations/Gaps**
Are Gans created equal? A large-scale study. (Lucic, Kurach, Michalski, Gelly, & Bousquet, 2018)	GANs	Celeba CIFAR-10 MNIST Digits MNIST Fashion	Frechet Inception Distance F1 Precision Recall	Most models can reach similar scores with enough hyper parameter optimization and random restarts.	There is still no clear consensus on which GAN model performs objectively better than others due to the computational budget to search over all hyperparameters.
Exploiting the generative adversarial framework for one-class multi-dimensional fault detection (Plakias & Boutalis, 2019)	GANs Autoencoders One-class SVM IF	Cardio Ionosphere Satellite	Accuracy of positive and negative samples	Introduce the use of a Generative Adversarial Network for one class fault detection, a demanding and urgent problem.	The fault detection rates of the GAN model are higher than that given by using the other algorithms.
Image super-resolution using progressive generative adversarial networks for medical image analysis (Mahapatra, Bozorgtabar, & Garnavi, 2019)	Progressive GANs	Mean-Square	Retinal MRI Cardiac MR	Proposed multi stage P-GAN outperforms competing methods and baseline GANs.	The saliency map-based approach had some limitations such as: 1) choice of optimal window size and weights for saliency map calculation depended on the specific image and was non-optimal.
HexaGAN: Generative Adversarial (Hwang, Jung, & Yoon, 2019)	Hexa-GAN	UCI MNIST	F1 Score RMSE	Apply suitable pre-processing techniques to the datasets.	The proposed framework is simple to use and works automatically when the absence of data.

(To be continued)

Related Work, Author, Year	Technique	Dataset	Metric	Results achieved	Limitations/Gaps
Improved techniques for training GANs (Salimans et al., 2016)	GANs	Test Error Rate. Inception Score	MNIST CIFAR-10 SHVN	Make the training of the GANs more stable and easier in the future.	Finding Nash equilibrium is a very difficult problem because the cost functions are non-convex, the parameters are continuous, and the parameter space is extremely high-dimensional.
Generative adversarial network in medical imaging: A review (Yi, Walia, & Babyn, 2019)	GANs			The recent work on styleGAN shows the capability to control the high-level attributes of the synthesized image.	The adoption of GANs in medical imaging is still in its infancy and there is currently no breakthrough application.
Improved Boundary Equilibrium Generative Adversarial Network (Li, Xiao, & Ouyang, 2018)	Boundary Equilibrium GANs	Inception Score	CIFAR-10 CelebA	The ability of discriminator in distinguishing between real and generated images is improved, which further guides the generator to produce more realistic images to confuse the discriminator.	The generated images look uniform with many noise-like regions, while at high values, the diversity of the generated images increases but the quality declines.
IntersectGAN: Learning Domain Intersection for Generating Images with Multiple Attributes (Yao et al., 2019)	Intersect GAN	Fréchet Inception Distance (FID)	Celeb Faces Attributes	Present a novel Intersect GAN framework to learn intersection of multiple image domains for generating image samples possessing multiple attributes without using real samples.	The proposed IntersectGAN is required to further extension to deal with more attributes with more scalable network architectures.

(To be continued)

Related Work, Author, Year	Technique	Dataset	Metric	Results achieved	Limitations/Gaps
Generative Adversarial Networks for Operational Scenario Planning of Renewable Energy Farms: A Study on Wind and Photovoltaic (Schreiber, Jessulat, & Sick,2019)	GANS DCGAN WGAN	KDE KLD	EuropeWindFarm2015& 2017 GermanSolarFarm2015& 2017	Provide a comparative study of two different loss functions (binary-cross-entropy loss and Wasserstein distance.	A critical remark of the analysis needs to be done concerning seasonal effects, which is challenging to consider due to the limited amount of data.
A Large Dimensional Study of Regularized Discriminant Analysis Classifiers (Elkhalil, Kammoun, Couillet, Al-Naffouri, & Alouini, 2017)		R-LDA R-QDA	USPS	Analyse and provide many results and corollaries using the Linear Discriminant.	The covariance matrix estimate becomes ill-conditioned or even non invertible, which leads to poor classification performance.
Forget the Learning Rate, Decay Loss (Jiakai Wei et al., 2018)	GANS	Mean-IOU	PSPNet	Prove the feasibility of gradient weight strategy in the field of computer vision.	Learning will become very slow despite the presence of a strong gradient because the learning rate must be shrunk to compensate for even stronger curvature.

From the extensive literature review of the previous related works that summarized in table 2.1. The researcher found that the main obstacles that hardens the development of GAN is the difficulty of training due to GAN highly sensitive to the hyperparameter selections. Also, Vanishing Gradient problem is one of the most phenomenal obstacles of GAN. Therefore, the researcher proposed a SGAN Classifier model for enhancing GANs networks' architectures that leads to better performance in terms of prediction accuracy as well as overcomes the limitations of existing works by designing the weight-optimized SCAN classifier model based on Population-Based Meta-Heuristic Sine Cosine Algorithm (SCA) instead of traditional manners.

2.7 Summary

Genetic algorithms identify the best candidate solution or the next best solution based on random initialization of several potential candidate solutions (Holland, 1992). Using the Sine-Cosine Algorithm to solve problems with unknown solution space would be simple. Furthermore, the Sine-Cosine Algorithm (Mirjalili, 2016) avoids local optima and coverage in the direction of the global optimum, focusing instead on exploring and exploiting different optimization regimes in the search space.

Due to its unique optimization concepts and theoretical advantages, the SCA algorithm has been proved to be superior to genetic algorithms, PSO, and other comparable approaches in the vast majority of situations; it has been used in a variety of practical applications (Somu, MR, & Ramamritham, 2020).

CHAPTER 3

METHODOLOGY

3.1 Overview

The main aim of this project is to build a single Fine-Tuned Generative Adversarial Network Classifier Architecture that would perform binary and multiclass Classification tasks efficiently, show the solution process as steps and how the selected Population-Based Meta-Heuristic Sine Cosine Algorithm (SCA) and other approaches implemented in the process of initialization of the parameters of the first layer that is the weights and the biases. After that, the researcher compares the results of the SCA applied model and the Baseline model to look for improvement in train and test accuracies.

3.2 The Methodology

The methodology used in this study is based on simulations and experiments methods which is the core part of this study. As can be seen, this thesis consists of two phases which show the sequence flow for every specific phase in the proposed methodology to achieve the targeted objectives of the project, the two phases are:

- **Phase One:** Survey and analysis of the state-of-the-art Generative Adversarial Networks classification model. This phase consists of two steps which are:
    I. Figure out the gaps and main challenges such as the effects and number of hyperparameters in Semi-supervised Generative Adversarial Networks.
    II. Identify the appropriate algorithm Sine Cosine Algorithm (SCA) and the four datasets (MNIST Digits, MNIST Fashion, Pneumonia Detection from Chest X-Ray images, and Facial Emotion Detection Dataset).
- **Phase Two:** Propose an Enhanced Weight Optimized Semi-Supervised Generative Adversarial Network Based on Sine Cosine Algorithm for Image Classification only. This phase consists of three iteration steps which are:
    I. Design and implement the proposed SGAN-SCA model.
    II. Validating the performance of the proposed model using four different benchmark datasets.
    III. Benchmark the results of the proposed method with the current state of the art.

Figure 3.1 clarifies the methodology of the proposed model.

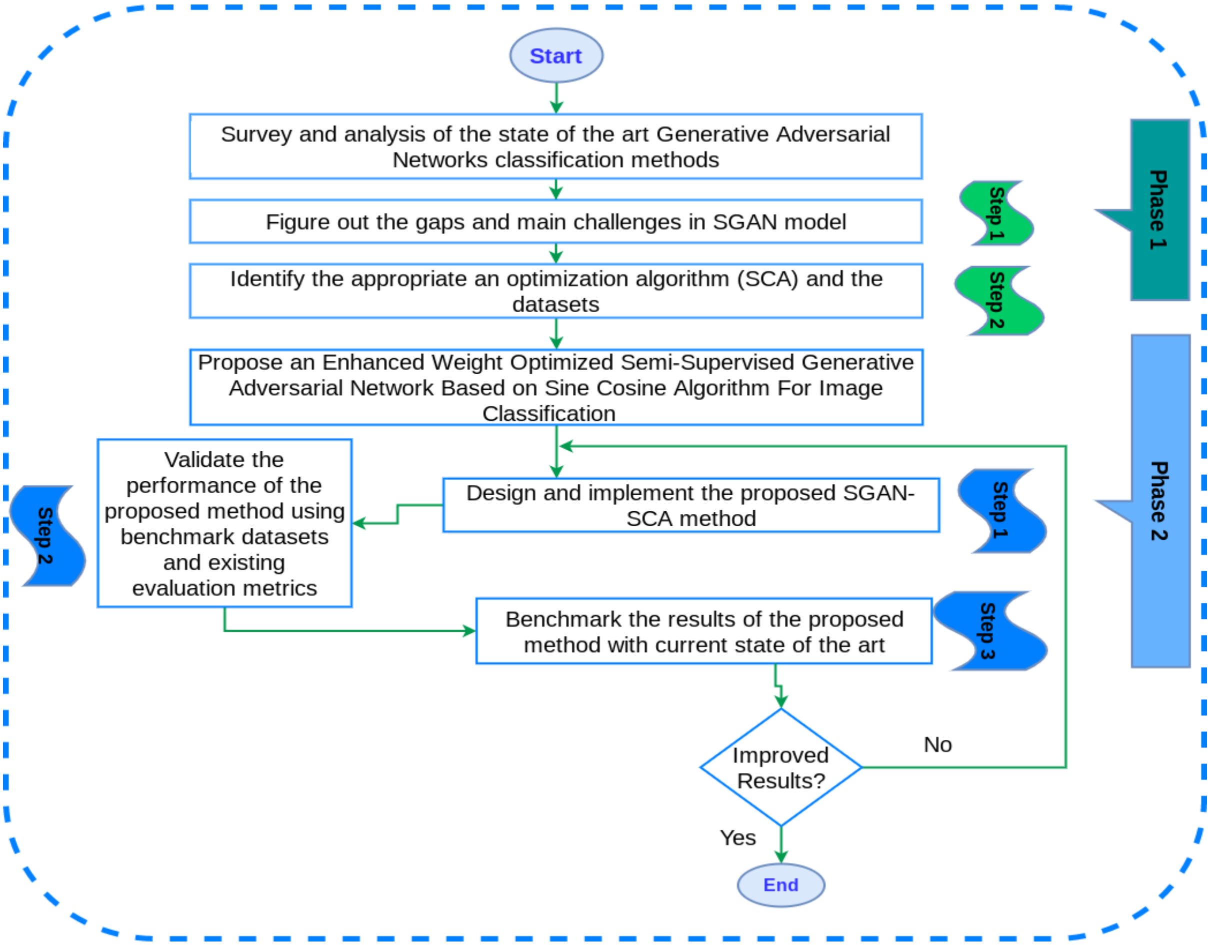

Figure 3.1: The methodology of the proposed model.

3.2.1 Literature Survey and Analysis

The survey and analysis in the first phase of the proposed methodology that was used to achieve the targeted objectives of this project. This phase has been illustrated in Chapter two by conducting an inclusive review of the machine learning section and its subsections such as supervised learning, semi-supervised, unsupervised learning, and reinforcement learning. Generative Adversarial Networks types have been reviewed deeply in chapter two as well, which is the main focus of this research to rely on, and these are the fundamental algorithms of deep learning. Moreover, the survey and analysis phase concentrates on identifying the starting of the study that is followed via survey and analysis of the state-of-the-art Generative Adversarial Networks classification models. Through this phase, the researcher has presented the survey and analysis techniques of the state-of-the-art approaches. The optimizing methods that focus on the weights parameters and generating hyper-parameters have been carefully studied and analyzed. The main use of these optimizing methods is adaptive to the dataset types and network structure.

This phase involves two steps which are to figure out the gaps and main challenges in Semi-supervised Generative Adversarial Networks as well as identify the appropriate algorithm Sine Cosine Algorithm (SCA) and the four multimedia benchmarking datasets (MNIST Digits, MNIST Fashion, Pneumonia Detection from Chest X-Ray images, and Facial Emotion Detection Dataset).

3.2.1.1 Gaps and Main challenges of GAN & SGAN.

The researcher found out that, the main obstacle which hardens the development of GAN is the difficulty of training due to being highly sensitive to the hyper parameter selections, and also vanishing gradient problem is one of the most phenomenal obstacles of the GAN model. Therefore, the researcher has designed and trained the typical GAN model to get deep knowledge about these obstacles and figure out the ultimate possible solutions which might participate to overcome these gaps and limitations. In consequence, the researcher proposed the SGAN model (Semi-Supervised GAN Classifier) to enhance GAN networks' architectures which will lead to better performance in terms of prediction accuracy by designing the weight-optimized SCAN classifier model based on the optimization algorithm instead of the traditional manner.

3.2.1.2Generative Adversarial Networks

The generative model is challenged against an opponent in the framework of the adversarial network: a discriminator model that learns to distinguish whether a sample is from the model distribution or the data distribution. The generative model can be compared to a group of counterfeiters attempting to create false currency and utilize it without being detected; in contrast, the discriminator model can be compared to the police attempting to find counterfeit currency. In this game, the competition forces both teams to improve their procedures until the counterfeits are indistinguishable from the real thing. This framework can produce tailored training algorithms for a wide range of models and optimization techniques (Goodfellow et al., 2014).

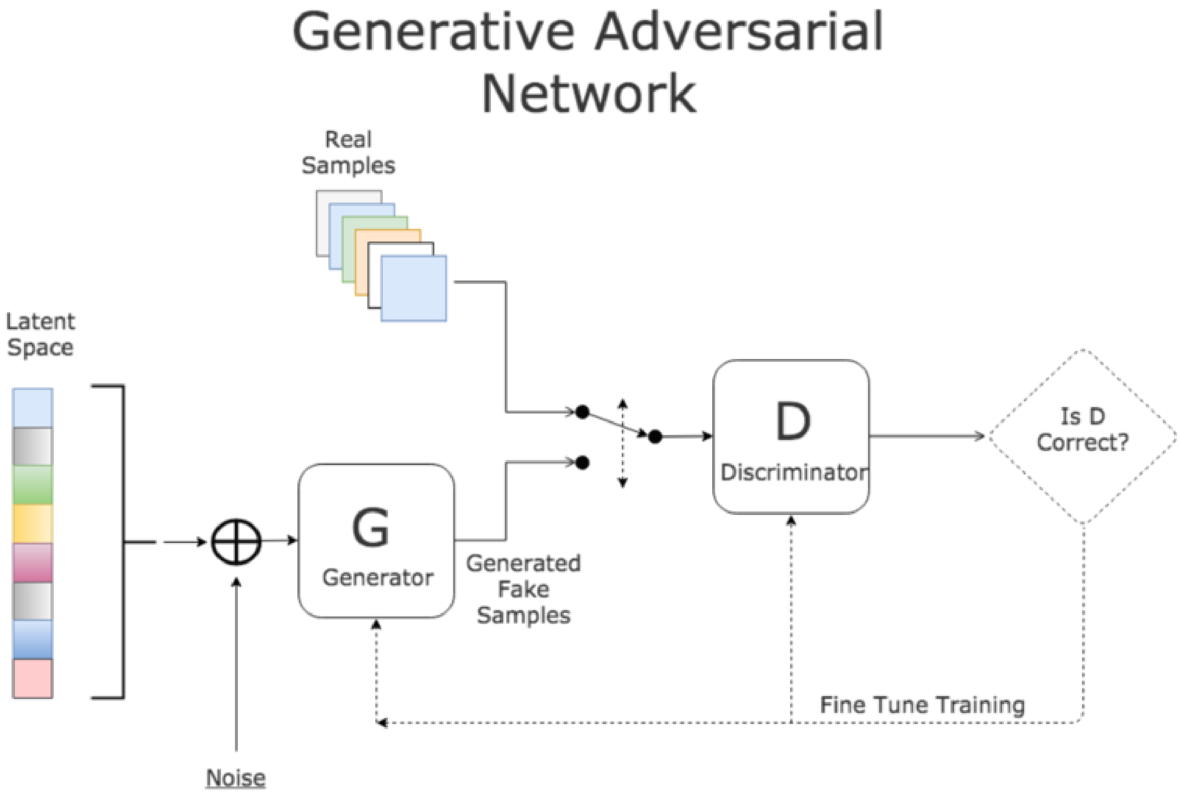

Figure 3.2: How the GANs work (Goodfellow et al., 2014).

When the models are both multilayer perceptrons, the adversarial modeling framework is rather simple to use. The researcher builds a prior on input noise variables Pz(Z) to learn the generator's distribution pg across data x, then represent a mapping to data space as G (Z; θg), where G is a differentiable function rep represented by a multilayer perceptron with parameters θg. A second multilayer perceptron D (x; θd) that outputs a single scalar is also defined. The likelihood that x came from the data rather than pg is represented by D(x). The researcher trains the D to assign the right label to both training examples and G samples with the highest likelihood whilst simultaneously training G to minimize the log-loss.

$$log\ log\ \left(1 - D\big(G(z)\big)\right) = V(D,G)\ = Ex - pdata(x)[logD(x)] +\ Ez - pz\left[log\ log\ \left(1 - D\big(G(z)\big)\right)\right] \tag{3.1}$$

As shown in figure 3.3, In the inner loop of training, optimizing D to completion is computationally costly, and on finite datasets, it would result in overfitting. Instead, the researcher alternates between k D optimization stages and one D optimization step. As long as G varies slowly enough, D will be maintained near its best solution. G may not be able to learn well if the gradient in the equation 3.1 is insufficient. G can reject samples with high confidence early in learning when G is weak because they are clearly distinct from the training data. log (1 −D(G(z))) saturates in this situation. The researcher can train G to maximize log(D(G(z))) instead of minimizing log (1 −D(G(z))). The fixed point of the dynamics of G and D is the same with this objective function, but the gradients are significantly larger early in learning (Goodfellow et al., 2014).

for number of training iterations **do**
 for k steps **do**
  • Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$

 end for
 • Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
 • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

end for

Figure 3.3: GAN Training Algorithm (Tim Salimans et al., 2016)

Hence, the gradient-based updates can use any standard gradient-based learning rule of the loss with respect to θd and θg can be computed with backpropagation because D and G are defined via well-understood neural network components. Here's the training algorithm from the previous GAN study (Goodfellow et al., 2014), and ideally once this is finished, pg=pdata, so G(z) will be able to produce new samples from pdata. The researcher used momentum in all experimental models.

$$\nabla\theta_d = \frac{1}{m}\sum_{i=1}^{m} [logD(x^{(i)}) + log(1 - D(G(z^{(i)})))]$$

$$\nabla\theta_g = \frac{1}{m}\sum_{i=1}^{m} log(1 - D(G(z^{(i)}))) \quad (3.2)$$

3.2.1.3 Semi-Supervised GAN Classifiers (SGANs)

This study aimed to build a single Fine-Tuned Generative Adversarial Network Classifier Architecture that can efficiently perform binary and multi-class classification tasks and apply population-based meta-heuristic sine cosine algorithm (SCA) to initialize the parameters of the first layer, namely the weights and the biases. After this, the researcher compared the results of the applied SCA model and the baseline model to look for improvements in train and test accuracies. Figure 3.4 shows the proposed SGAN diagram of this study.

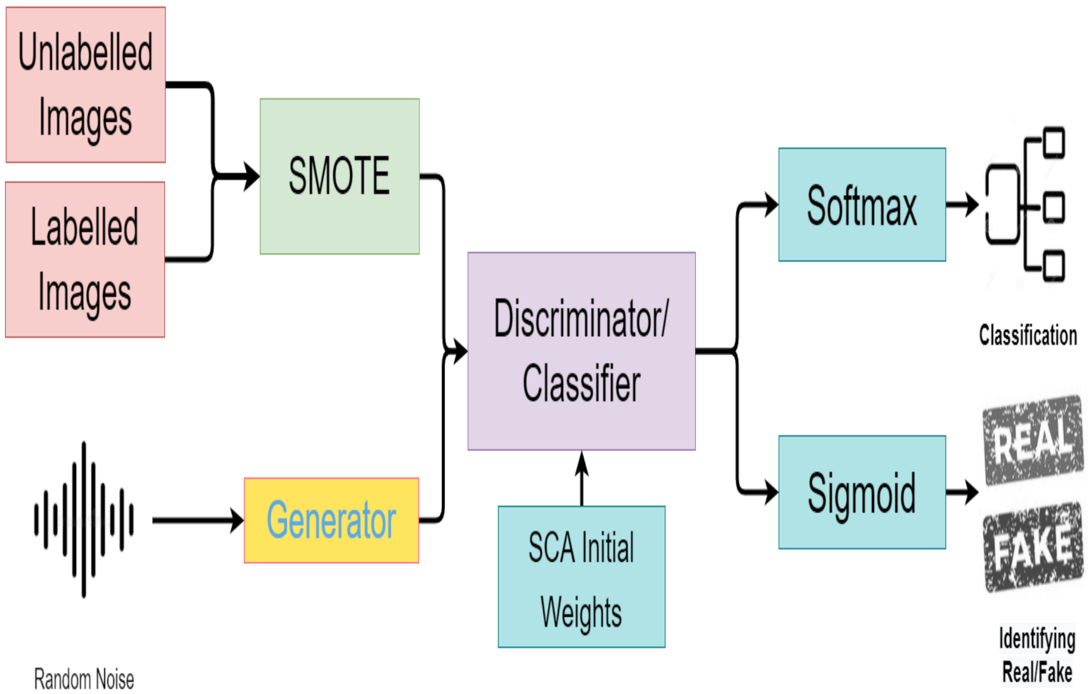

Figure 3.4: The proposed SGAN diagram.

In a normal SGAN, a generative net G and a discriminator D are trained simultaneously with conflicting objectives. The discriminator network D outputs a probability that the input image is drawn from the data generating distribution. A generative model for images while the researcher trained an image classifier will be called G. Typically, a feed-forward network with a single sigmoid unit is used, but it can alternatively be done with a softmax output layer with one unit for each of the classes [REAL, FAKE]. D may have N+1 output units matching [CLASS-1, CLASS-2,... CLASS-N, REAL/FAKE] after this change is done. In this situation, D can also take on the role of C. This network is referred to as D/C. It's comparable to training a GAN to train a SGAN. The researcher simply applied higher granularity labels for half of the minibatch drawn from the data generating distribution. G is taught to maximize the negative log probability with regard to the supplied labels, whereas D/C is trained to decrease it concerning the given labels (Salimans et al., 2016).

Algorithm 1 SGAN Training Algorithm

Input: I: number of total iterations

for $i = 1$ **to** I **do**

Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

Draw m examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ from data generating distribution $p_d(x)$.

Perform gradient descent on the parameters of D w.r.t. the NLL of D/C's outputs on the combined minibatch of size $2m$.

Draw m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.

Perform gradient descent on the parameters of G w.r.t. the NLL of D/C's outputs on the minibatch of size m.

end for

Figure 3.5: SGAN training algorithm (Tim Salimans et al., 2016).

Hence, for the discriminator training, the data label coming from $P_{data}\,(x)$ is $y = 1$ (real data) and $y\hat{} = D(x)$. So, the discriminator loss is given in eq (3.3).

$$L(D(x), 1) = log\ log\ (D(x)) \tag{3.3}$$

and for data coming from the generator, the label is $y = 0$ (fake data) and $y\hat{} = D(G(z))$.Thus, in this case,

$$L(D(G(z)), 0) = log\ log\ (1 - D(G(z))) \tag{3.4}$$

Ultimately, the discriminator's purpose is to classify the fake and real datasets appropriately. To do this, eq. (3.3) and eq. (3.4) should be maximized, and the discriminator's final loss function can be written as in eq. (3.5).

$$L^{(D)} = \left[log\ log\ (D(x)) + log\ log\ (1 - D(G(z)))\right] \tag{3.5}$$

In this case, the generator faces off against the discriminator. As a result, it will attempt to minimize eq. (3.5), and the loss function is specified as,

$$L^{(G)} = \left[log\ log\ (D(x)) + log\ log\ (1 - D(G(z)))\right] \tag{3.6}$$

Therefore, by combining eq (3.5) and eq (3.6) considering the whole dataset, the final loss function will be as

$$\underset{G}{min}\underset{D}{max}V(D,G) = \underset{G}{min}\underset{D}{max}\left(E_{x\sim P_{data}(x)}[log\ log\ D(x)\ ] + E_{z\sim P_z(z)}\left[log\ log\ \left(1 - D\big(G(z)\big)\right)\right]\right) \tag{3.7}$$

Apart from that, table 3.1 illustrates the difference between generator and discriminator in the proposed SGAN model.

Table 3. 1: Difference between Generator and Discriminator

Category	**Generator**	**Discriminator**
Input Data	Vector of random values(pixel values) of the size of the image sampled from the normal distribution	Three kinds of inputs: real unlabeled samples(from the dataset), labelled real samples (from the dataset), and fake samples (from the generator)
Output Data	Fake Samples that try to imitate the real samples as closely as possible	Softmax probabilities of the different classes of the output and prediction of the image being real or fake
Target	Generate fake samples by taking references from the real images and trying to make them indistinguishable from the real images from the dataset through training of the weights and biases	Try to correctly discriminate between real and fake images and assign correct class labels to each input, be it real or fake.

3.2.1.4 Sine Cosine Algorithm (SCA)

SCA is a recent effective optimization algorithm capable of exposing an efficient performance and is more effective than several optimization algorithms for achieving an optimum or near optimum solution; as a result, it was chosen for this study due to its unique characteristics as stated in the literature and motivation section (Somu & Ramamritham, 2020). The researcher used SCA to enhance models in two different scenarios, namely SGAN-SCA and SMOTE-SGAN-SCA.

Sine-Cosine Algorithm is a recent advancement in Metaheuristic Population-Based Optimization algorithms. It was proposed by Seyedali Mirjalili (Mirjalili, 2016).

$$X_i^{t+1} = X_i^t + r_1 * Sin\ (r_2) * |r_3P_i^t - X_i^t|\ X_i^{t+1} = X_i^t + r_1 * Cos\ (r_2) * |r_3P_i^t - X_i^t| \tag{3.8}$$

As is common to algorithms belonging to the same family, the optimization process consists of the movement of the individuals of the population within the search space, which represent approximations to the problem. For this purpose, SCA uses trigonometric sine and cosine functions. At each step of the calculation, it updates the solutions according to the following equations:

Where X_{ti} is the position of the current candidate at the tenth iteration in the ith dimension, Pi is the position of the best candidate at the tenth iteration in the ith dimension. The random agents are r_1, r_2, r_3, and r_4.

The equations can be combined and written together as follows:

$$X_i^{t+1} = \{X_i^t + r_1 * Sin\ (r_2) * \left|r_3 P_i^t - X_i^t\right|, r_4 < 0.5\ X_i^t + r_1 * Cos\ (r_2) * \left|r_3 P_i^t - X_i^t\right|, r_4 \geq 0.5$$

(3.9)

The first agent (r1) defines the subsequent search space, located between the solution region or outside it.

The second operator (r2) defines the distance in the search space that should be in or out of the destination.

$$r_i = a - t\frac{a}{T} \tag{3.10}$$

Where T indicates the maximum iterations number and t is the currently running iteration. A is a constant variable.

After all, the pseudo code of the SCA algorithm is presented in figure 3.6 where the SCA algorithm starts the optimization process by a set of random solutions. The algorithm then saves the best solutions obtained so far, assigns it as the destination point, and updates other solutions with respect to it (Mirjalili, 2016)..

Algorithm 1 The pseudo-code of the Sine Cosine Algorithm

:

Initialize a set of solutions $X_i (i = 1, 2, \ldots, n)$ randomly
while t is less than T_{max} **do**
    Calculate the objective value for each solution
    Update the destination $(P = X)$
    Update the random parameters r_1, r_2, r_3, and r_4
    Update the solutions using equation (3)
end while
Return the destination P

Figure 3. 6: General steps of the SCA Algorithm(Mirjalili, 2016).

SCA has been found more efficient than other population-based algorithms in achieving an optimal global solution (Somu & Ramamritham, 2020). Several search areas are investigated when the sine and cosine functions return values greater than one or less than one.

3.2.1.5 Multimedia Benchmark Datasets

The researcher primarily benchmarked all the experimental models with four datasets which are all image-based classification datasets. These datasets are:

1. **MNIST Digits Dataset -** contains handwritten examples of digits from 0-9 with the labels corresponding to each drawing showing the digit it represents (Deng, 2012).

Figure 3. 7: MNIST Digits Dataset (Deng, 2012).

MNIST Digits is a dataset consisting of a training set of 60,000 examples and a test set of 10,000 examples (Deng, 2012). Each example is a 28x28 grayscale image associated with a label from 10 classes. Each image is 28 pixels in height and 28 pixels in width, for 784 pixels in total. Each pixel has a single pixel value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see above) and represents the digits from 0 to 9. The rest of the columns contain the pixel-values of the associated image. Each training and test example is assigned to one of the following labels: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

2. **MNIST Fashion Dataset** - contains examples of various garments (10 types) like - shirts, pants, boots, headwear, etc.; and their corresponding labels (Xiao, Rasul, & Vollgraf, 2017).

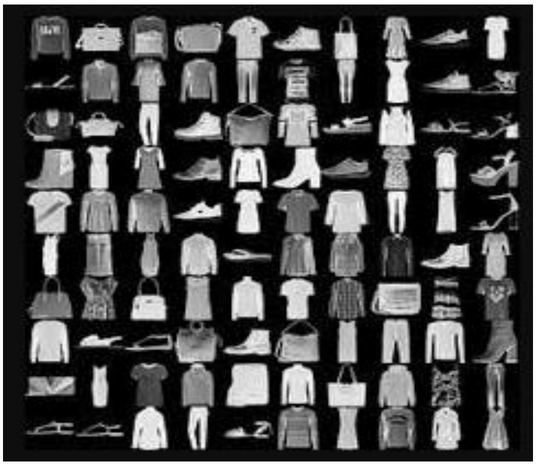

Figure 3. 8: MNIST Fashion Dataset (Xiao, Rasul, & Vollgraf, 2017).

Fashion-MNIST is a dataset consisting of a training set of 60,000 examples and a test set of 10,000 examples (Xiao, Rasul, & Vollgraf, 2017). Each example is a 28x28 grayscale image associated with a label from 10 classes. Fashion-MNIST is intended to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits. Each image is 28 pixels in height and 28 pixels in width, for 784 pixels in total. Each pixel has a single pixel value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see above) and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image. Each training and test example is assigned to one of the following labels: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot.

3. **Pneumonia Detection from Chest X-Rays** - this dataset contains two classes, chest X-Ray images of normal vs. pneumonia lungs (Mooney, 2018).

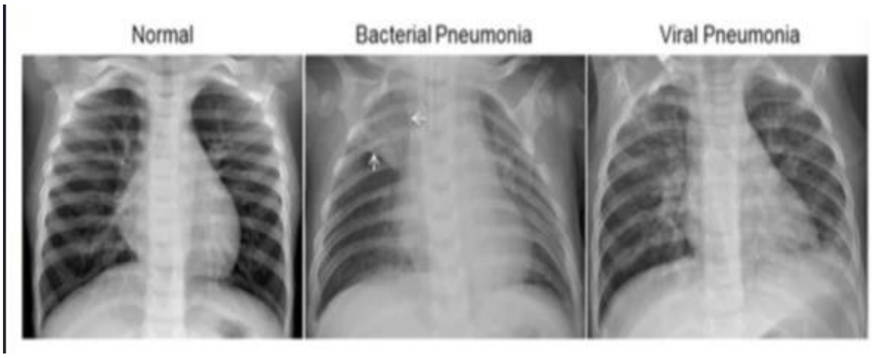

Figure 3. 9: Pneumonia Chest X-Ray Dataset (Mooney, 2018).

The Pneumonia Chest X-Ray images dataset (Mooney, 2018). The dataset is organized into three folders (train, test, Val) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and two categories (Pneumonia/Normal). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low-quality or unreadable scans. Two expert physicians then graded the diagnoses for the images before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert. The normal chest X-ray depicts clear lungs without any areas of abnormal opacification in the image. Bacterial pneumonia typically exhibits a focal lobar consolidation, in this case in the right upper lobe, whereas viral pneumonia manifests with a more diffuse" interstitial" pattern in both lungs. For the task at hand, the researcher preprocessed the image dataset into the table of pixel values where the table contains 784(28×28) columns, each containing the pixel values of the images and another column denoting the class label of the image categories - 0 for Normal and 1 for Pneumonia.

4. **Facial Emotion Detection Dataset** - this is a tricky dataset with large size images of 7 different classes of facial emotions (Sambare, 2021).

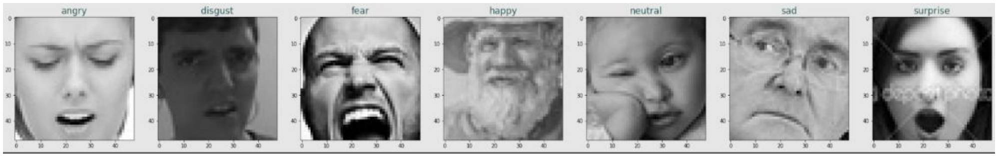

Figure 3. 10: Facial Emotion Recognition Dataset (Sambare, 2021).

The Facial Emotion Recognition 2013 Dataset consists of 48x48 pixel RGB images of faces (Sambare, 2021). For the present analysis, the images have been grayscale. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples, and the public test set consists of 3,589 examples. Just like the previous example, the researcher converted the images to the tables of pixel values such that the table contains a total of 2305 columns - 2304(48 x 48) columns representing the pixel values extracted in a row-major fashion and 1 column depicting the class/category the image belongs to. In all, there are 32298 attributes in the tabular dataset.

3.2.2 Enhanced Weight Optimized SGANs using Sine Cosine Algorithm

3.2.2.1 Design Original GAN

The researcher has developed a typical and full GAN model using a python environment for generating images. The Original GAN includes a discriminator model for classifying whether a given image is real or generated and a generator model that uses inverse convolutional layers to transform an input to a full two-dimensional image of pixel values. It can be challenging to understand both how GANs work and how deep convolutional neural network models can be trained in a GAN architecture for image generation. Therefore, the researcher trained the original GAN via using small and well-understood dataset (MNIST handwritten digit dataset) as the smaller model can be developed, trained quickly and allowing the focus to be put on the model architecture and image generation process itself. During the training process of the original GAN, the researcher used the grid search to adjust the model's parameters. The goal would be to minimize the cost function so that the model learns the probability distribution of the output given the input. After the training phase, the researcher used the model to classify a new handwritten digit image by estimating the most probable digit the input corresponds.

3.2.2.2 Design Baseline Models (SGAN Classifier)

The researcher then developed an SGAN classifier, following the instructions mentioned in the study by Tim Salimans et al. (2016). This is the project baseline model where the researcher fine-tuned the hyperparameters like the learning rates and the batch sizes for the SGAN model by manual searching.

Initially, the data was preprocessed to get the images in the form of a pixel value table with each of the scaled pixel values as the features and the target as the labels of the images. The generator model was next defined, followed by the discriminator and classifier models. In a broad sense, the classifier model is similar to the discriminator model; the only thing that is different in-between them is the output layer. The discriminator has a sigmoid layer for classifying real/fake images, and the classifier has a softmax layer used to perform the multiclass classification. The baseline models were trained for classification tasks on all the benchmarking datasets. It was found that the datasets 3 and 4 had class imbalances leading to poorer results than expected. So, the researcher decided to try two techniques to deal with the problem - (i) Synthetic Minority Oversampling Technique (SMOTE) (ii) Principal Component Analysis(PCA).

3.2.2.2.1 SMOTE

SMOTE was applied on all four datasets before feeding them into the model and then trained the SGAN model. The researcher found the results as expected since datasets 1 and 2 were balanced, the results remained almost the same as the baseline results. For the unbalanced datasets 3 and 4 results greatly improved in the accuracy values after applying SMOTE technique. In SMOTE, the” no majority” sampling strategy was used to resamples all classes. Still, the majority class and nearest neighbors set the default value of 5, which was used to construct the synthetic samples in the datasets.

3.2.2.2.2 PCA

The researcher also tried applying PCA on the datasets for dimensionality reduction, but the idea didn’t lead to any improvement in the accuracy scores. Furthermore, reduction dimensions’ hyperparameter tuning also didn’t significantly change scores, so the researcher discarded the idea. PCA was implemented with an “auto” SVD solver. The solver is selected by a default policy based on X shape and n components. If the input data is larger than 500x500 and the number of components to extract is lower than 80% of the smallest dimension of the data, then the more efficient ‘randomized’ method is enabled. Otherwise, the exact full SVD is computed and optionally truncated afterward.

3.2.2.3 Optimizing The proposed model (SGAN-SCA)

Weight initialization is an important consideration in the design of the SGAN model. The nodes in the SGAN model are composed of parameters referred to as weights used to calculate a weighted sum of the inputs (Brownlee, 2021). The SGAN model is fitted using an optimization algorithm called the Sine Cosine algorithm(SCA) that incrementally changes the network weights to minimize a loss function, hopefully resulting in a set of weights for the mode that is capable of making useful predictions.

SCA has been found more efficient than other population-based algorithms in achieving an optimal global solution (Somu & Ramamritham, 2020). Several search areas are investigated when the sine and cosine functions return values greater than one or less than one. The SCA requires a starting point in the space of possible weight values from which to begin the optimization process. Weight initialization is a procedure to set the weights of the SGAN model to small random values that define the starting point for the optimization (learning or training & Batch size) of the SGAN model. GS is efficiently applied for each experimental setup model to find the best optimal set of the hyperparameters (Learning rate & Batch size) by locating and picking the most appropriate values for the dataset size and SGAN model structure. Therefore, the researcher defined the optimal learning rate sets between decay values 1e-4 and 2e-4 to avoid the vanishing and exploding gradient problem.

The figure 3.11 summarized the data preparation and preprocessing of the proposed model where the data preparation is the process of cleaning and transforming raw data before processing and interpretation. The termination condition has been used to check the final result of the SGAN model after the SCA algorithm is applied for initializing the best optimal set of the hyperparameters (weights and biases) in the discriminator and classifier models to initialize the first layer's parameters (2D Convolutional Layer) instead of randomly initializing the weights with a SEED of 3.

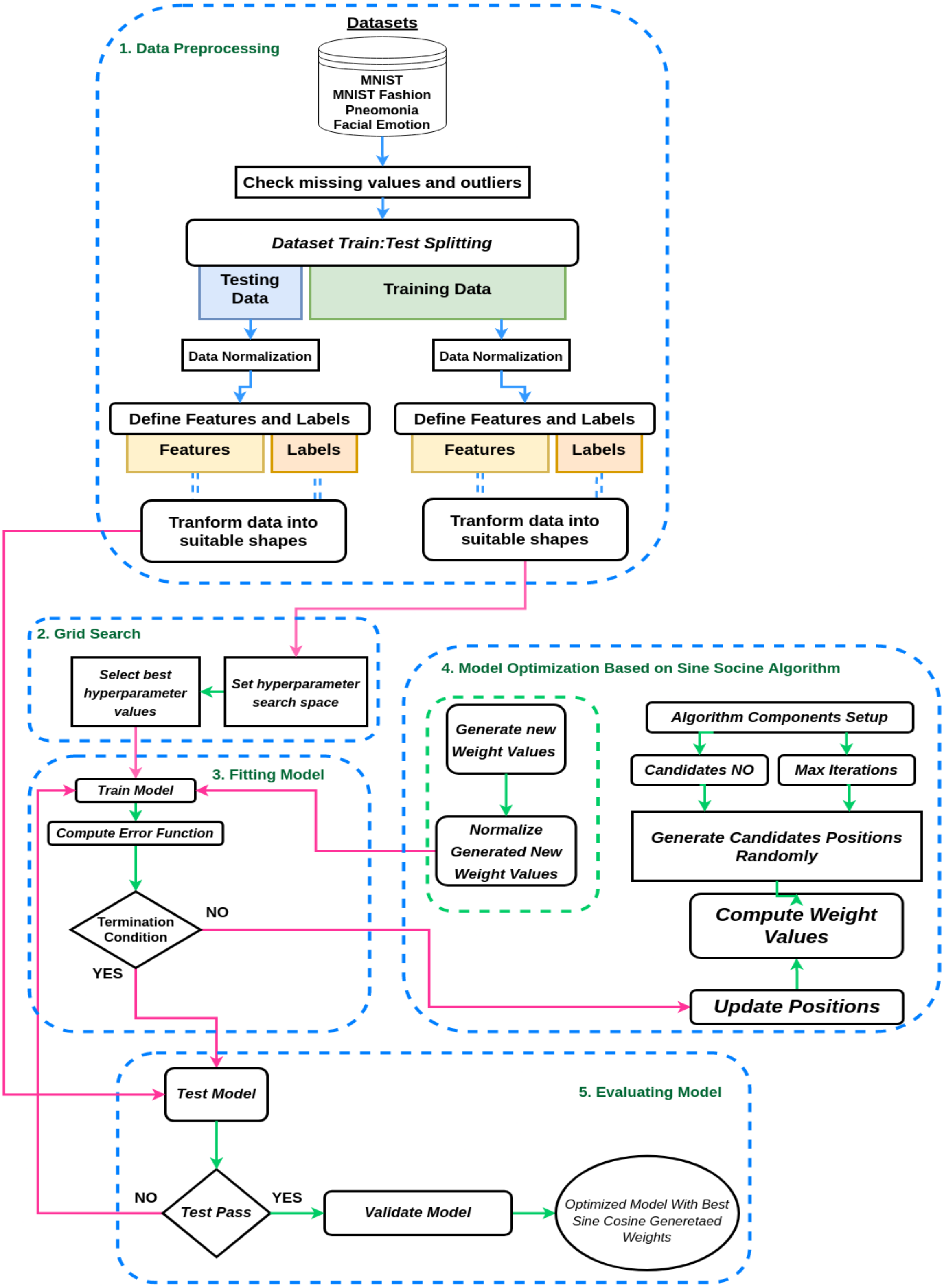

Figure 3. 11: General model of the proposal SGAN-SCA.

After all these were done, the researcher finally proceeded to the last part of the project, i.e., a script was built for applying Sine-Cosine Algorithm for weights and bias initialization in the discriminator and the classifier models. The researcher could not do the same thing for the generator model because the first layer of the generator model was a Dense/Fully Connected Layer. As a result, there are too many parameters, such that it's practically impossible to train them all. After training all the combinations of the models, the researcher trained them on all the four benchmarking datasets to get expected improvements upon using the Sine Cosine Algorithm to initialize the Parameters instead of doing it randomly.

3.3 Computing Environment

The hardware platform utilized in this project is the Intel(R) Core™ machine equipped with i7-8700, running at speed 3.7 GHz CPU and 16 GB of RAM. The software used for implementing the proposed method is python 3.7.8. The software environment used to conduct the experiments was Python 3 programming language and the libraries below:

- Environment setup:
    - Python version (Python 3.7.8)
    - Virtual environment from Anaconda
    - TensorFlow (2.6.0) as backend.
- Library setup:
    - Scikit-learn (0.23.2)
    - Scipy (1.4.1)
    - Pandas (1.1.3)
    - Numpy (1.18.5)
    - Matplotlib (3.3.2)
    - Seaborn (0.11.1)
    - Keras (2.4.3)

3.4 Data Preparation and Preprocessing

Data preparation is the process of cleaning and transforming raw data before processing and interpretation. This is a crucial stage before processing, and it usually includes reformatting the data, correcting it, and integrating data sets to enrich it. Data processing is often a time-consuming operation for data professionals or corporate users, but it is a necessary precondition for observations and the elimination of partialities caused by insufficient data consistency. During the data preparation phase, for example, standardizing data formats, enhancing source data, and/or removing outliers are common.

3.4.1 Read and Discover Data

Understanding data before dealing with it is not simply a good idea; it's necessary to deal with long-term results. Data discovery may be performed through the use of summary statistics and visualizations to better evaluate data and uncover suggestions about the data's patterns consistency and to communicate findings and the intended study's hypothesis. The key concept is that the researcher must first evaluate the natuproperly picking prediction algorithms or arranging the next data preparation processes of data preparation. Some of the datasets were taken from open-source resources such as TensorFlow datasets in the case of mnist digits and fashion mnist datasets. The others were taken from the Kaggle datasets repository.

3.4.2 Cleaning and Validating Data

Cleaning up data has traditionally been the most time-consuming part of the data preparation process, but it is critical to eliminate inaccurate data, fill in missing values, and validate that the data follows defined patterns. After the data has been cleared, it must be double-checked for any problems that have been detected during the data processing step. During this stage, any flaws in the procedure are usually discovered and must be addressed before proceeding. The cleaning and validation of the data are explained in the following subsections.

3.4.2.1 Missing Values

In a dataset, missing values are either empty or have no value assigned to them. These fields are frequently seen when merging many columns from different databases, and they are usually the consequence of data input mistakes during the data collecting process. Dealing with nan values can be done in a variety of ways. The easiest technique is to use the mean, median, mode, or constant to fill in missing numbers. Another option is to delete records from the data collection directly. The researcher used the KNN imputer method from sci-kit-learn, which takes the mean of the five surrounding cells by default. For continuous data, the alternative option uses Euclidean distance metrics. The KNN method is straightforward to comprehend and implement. It also keeps the data distribution since it only takes the mean of five neighboring data but not the whole column.

3.4.2.2 Data Normalization

Normalizing information is the demonstration of changing the configuration or trait sections to accomplish an obvious outcome or to make the information more reasonable to a more extensive crowd. It is the most common way of adding and connecting information with other pertinent data to give significant viewpoints. The researcher utilized the MinMaxScaler procedure from the scikit learn python library to change the information into the state of (0,1).

3.4.2.3 Split Data

The data has been partitioned into 80% for training data, 10% for validation data, and the rest 10% as testing data. The splitting was done with the equal ratio of target label classes split for all three data. This was done using the stratify option of sci-kit-learn train_test_split sub-option from the model selection option (Cerqueira, Torgo, Smailović, & Mozetič, 2017; Bergmeir, Hyndman, & Koo, 2018).

3.4.3 Grid Search

Deep learning models need many parameters to be established before training to deliver reliable results (Cherkassky & Ma, 2002; Hsu, Chang, & Lin, 2003). These hyperparameters must be appropriate for the network topology and dataset type. These hyperparameters can be manually set by trial and error until the best results are obtained; however, this is an inefficient technique because it takes time and may not provide the desired results (Cho et al., 2014; Feng, 2011). As a result, numerous hyperparameter tuning strategies have automatically created models to make them suited for the network's topology (Hinz, Navarro-Guerrero, Magg, & Wermter, 2018; Shuai, Zheng, & Huang, 2018). Grid Search is a hyperparameter tuning approach that may be used in a variety of situations to find the best model configuration (Ragab, Abdulkadir, Aziz, et al., 2020); it produces more accurate results (Hinz, Navarro-Guerrero, Magg, & Wermter, 2018; Ragab, Abdulkadir, & Aziz, 2020). Therefore, GS has been utilized to obtain the best learning rate values in this thesis.

In order for a neural network to perform well, it must have the best hyperparameters. Trial and error have traditionally been employed to determine the optimum hyperparameters; however, this method is ineffective since it takes a long time and, in most circumstances, the optimal values are not guaranteed. It must attempt to specify the parameters, train the model, validate the model, test it, and compute the error. This cycle procedure must be done several times before it starts to yield improved results. As a result, the researcher employs a powerful approach known as grid search to quickly identify the optimum hyperparameters for the models under consideration.

Table 3.2 displays the values of the search space that the GS uses to locate and pick the most appropriate values for the dataset size and model structure. The optimal learning rate should not be too small or too big to avoid the vanishing and exploding gradient problem.

Therefore, The researcher set the search space to be between 1e-4 and 2e-4. Table 3.3 illustrates the optimal hyperparameters of all the experimental models.

Table 3. 2: Grid Search Hyperparameters Parameters Search Space

Grid Search Parameters	**Initialization Values**
Learning Rate	[1e-4, 2e-4]
Dropout Rate	[0.1, 0.2,0.4]
Batch Size	[128, 256]

Table 3. 3: Optimal Hyperparameters

Dataset	**Learning Rate**	**Batch Size**
MNIST Digits	2e-4	128
MNIST Fashion	1e-4	256
Pneumonia Chest X-Ray	2e-4	128
Facial Emotion Detection	2e-4	256

3.4.4 Evaluation Measures

In this study, four common metrics are used to thoroughly evaluate the effectiveness and prediction of the proposed models: Accuracy, Precision, Recall, and F1-Score. The following are the definitions of these four metrics. As the name suggests, Confusion Matrix gives the researcher a matrix as output and describes the complete performance of the model.

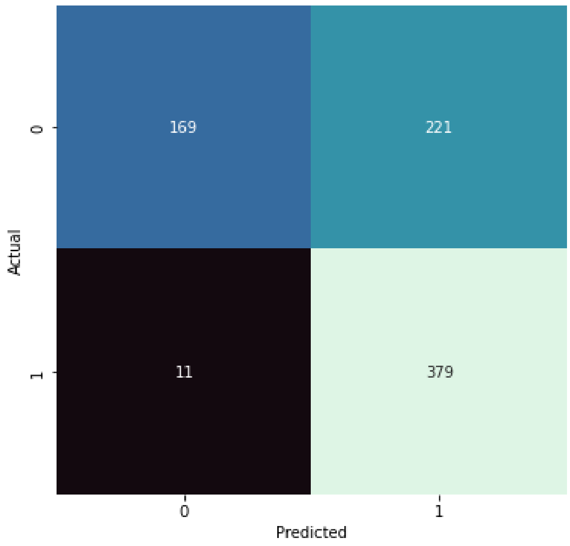

Figure 3. 12: An Example Confusion Matrix

There are 4 important terms (Agarwal, 2020):

1. True Positives: The cases in which the proposed model predicted YES (1) and the actual output was also YES (1). (i.e. 379 in the figure above)
2. True Negatives: The cases in which the proposed model predicted NO (0) and the actual output was NO (0). (i.e. 169 in the figure above)
3. False Positives: The cases in which the proposed model predicted YES (1) and the actual output was NO (0). (i.e. 221 in the figure above)
4. False Negatives: The cases in which the proposed model predicted NO (0) and the actual output was YES (1). (i.e. 11 in the figure above)

Accuracy (Agarwal, 2020) is the ratio of the number of correct predictions to the total number of input samples, i.e.

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample} \tag{3.11}$$

Precision (Agarwal, 2020) is the ratio of the true positives predicted by the proposed model to the sum total of the labels predicted as true positives and false positives.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3.12)$$

Recall is the ratio of the true positives predicted by the proposed model to the sum total of the labels predicted as true positives and false negatives (Agarwal, 2020).

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3.13)$$

F1-Score (Agarwal, 2020) is the Harmonic Mean between precision and recall, and hence the value of the F1 score is always a real number between 0 and 1. This value captures how precise and accurate the model is in classifying the output classes for the labels.

$$F1 - Score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \quad (3.14)$$

3.5 Summary

This chapter serves as the research methodology followed in this thesis to achieve the targeted objectives. The chapter showed the two phases of methodology and the flowchart, which explained the sequence flow of the stages that are followed in the methodology to achieve the thesis objectives. The flowchart of the methodology contains two phases. The first phase, covered in Chapter 2, is the analysis stage, which is a survey of previous related work in optimizing recurrent neural networks. The second phase, covered in Chapter 3, presents the proposed weight-optimized method by implementing the SCA algorithm. Then, discusses the experimental setup as well as the datasets and evaluation measures that are used for benchmarking purposes. Besides, the proposed enhanced method is benchmarked with state-of-the-art algorithms using the four datasets and evaluation measures mentioned.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview

This chapter demonstrates and discusses the performances of the proposed classification of images using SGANs based on the Sine Cosine Algorithm(SCA) for Predictive Analytics. After the researcher selected the environment setup of the experiment's components and identified the multimedia benchmarking datasets as well as evaluation classification metrics for the model validation purpose. All the experimental models have been run many times for each database to ensure the stability of the proposed models, the models have been run many times for each dataset. The results of all the experimental models have been discussed and then compared to each other based on the train and test accuracy of the selected multimedia benchmarking datasets. Lastly, the results of the proposed model have been compared with different existing deep learning methods that have been used to perform classification in the literature

4.1 The experimental models Results.

Through this section, the researcher is going to give a detailed explanation on how the experimental models have been conducted, trained, tested and then the results of these models are presented in tables and a comprehensive discussion of these results will be conducted as well.

4.1.1 Results of Original GAN Model

The researcher has designed the Original GAN model through a typical and full GAN model using a python environment and then trained the generative model and discriminator on a MNIST dataset with inputs belonging to one of N classes. The model improvement in generation of the images over iterations is very obvious and it would be best to compare the results between the 0& the 9800 iterations.

The figures 4.1& 4.2 illustrate how the GAN model generates the images after each iteration. The model improvement can be seen very clearly in the images after each iteration(epoch). Figure 4.1 is the first image that generated by the model after training for one epoch. Clearly, nothing is there except random noise. Aside of that, the figure 4.2 is the last image generated by the model after training for 9800 epochs. This image looks promising and a lot better than the adjacent one.

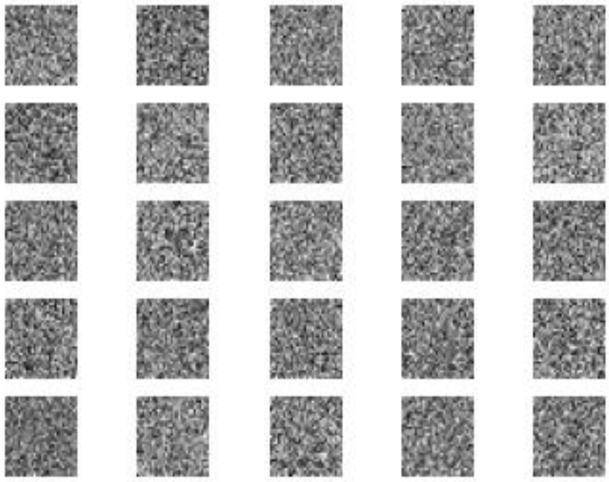

Figure 4.1 : MNIST- Image No.0

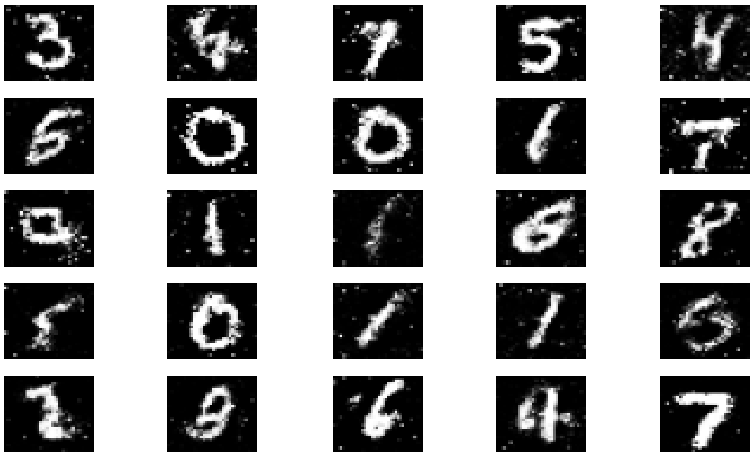

Figure 4.2: MNIST- image No.9800

Figure 4.3 is the GAN model heat map of the no. of samples with which digit they actually were on the Y-axis and what they were predicted by the GAN model.

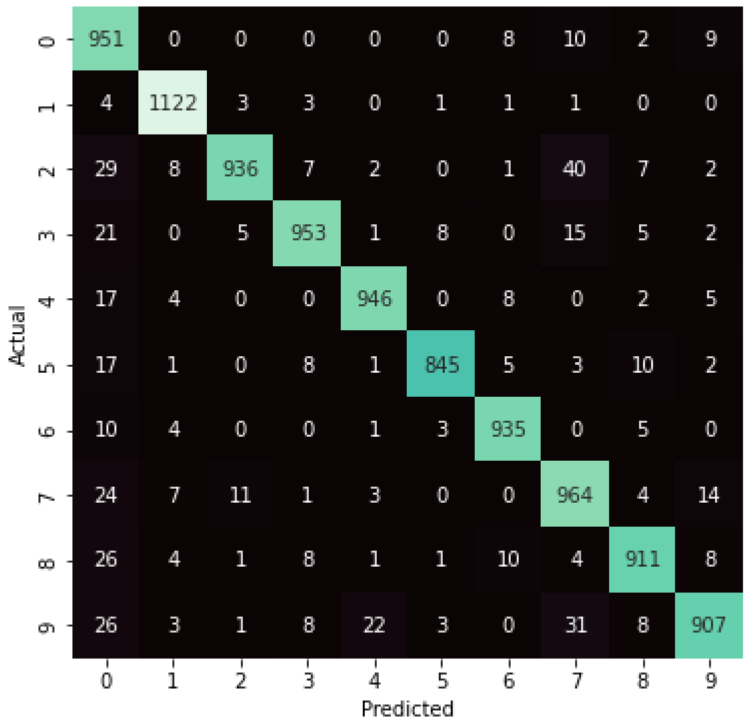

Figure 4.3: GAN Confusing Matrix Heat Map.

Table 4.1 presents the classification results of the original GAN model that trained in the MNIST dataset where these results were very strong with 95% in the accuracy, precision, recall and F1-score respectively. After that, the researcher designed the SGAN classifier model as an extension of the Generative Adversarial Network architecture for addressing semi-supervised learning problems.

Table 4.1: Original GAN classification Result.

Dataset	**Learning Rate**	**Batch Size**	**Accuracy**	**Precision**	**Recall**	**F1-score**
MNIST	2e-4	128	95%	0.95	0.95	0.95

4.1.2 Results of Baseline Model(SGAN).

The primary goal of developing the Baseline mode (SGAN) is to improve the effectiveness of generative adversarial networks for semi-supervised learning. Therefore, the Baseline results consist of the classification results on the four datasets whereby the model is a simple Semi-Supervised Generative Adversarial Network classifier that is directly trained and tested on the pixel value tables of the dataset's images. Table 4.2 illustrates the results of the Baseline SGAN model where these results prove that benchmark datasets like the MNIST digits dataset shows superior results under Baseline conditions, majorly due to how easy it is to classify handwritten digits, given how complex and fine-tuned the SGAN classifier is. However, when the difficult datasets are taken like the Facial Emotion Detection dataset wherein the predicted facial emotion of a person from the image of the face provided, the same complex SGAN classifier becomes counterproductive, leading to inaccurate results in most cases. As it is known that GANs are difficult to train, and it's difficult to achieve good results at times.

Finally, it found that the datasets Pneumonia Chest X-Ray and Facial Emotion Detection had class imbalances leading to poorer results than expected. So, the researcher decided to use the Synthetic Minority Oversampling Technique (SMOTE) for overcoming this problem.

Table 4. 2: Results of the Baseline SGAN Model

Dataset	**Learning Rate**	**Batch Size**	**Train Acc.(in %)**	**Test Acc.(in %)**	**Precision**	**Recall**	**F1-score**
MNIST Digits	2e-4	128	93.48	94.12	0.93	0.93	0.93
MNIST Fashion	1e-4	256	74.11	73.75	0.83	0.73	0.73
Pneumonia Chest X-ray	2e-4	128	74.77	63.14	0.81	0.81	0.63
Facial Emotion Detection	2e-4	256	25.13	24.72	0.26	0.21	0.23

4.2.3 Results after Balancing Datasets

The researcher applied the SMOTE technique with the SGAN classifier model to deal with the class imbalances problem in the unbalanced datasets like Pneumonia Detection from Chest X-Ray Images and Facial Emotion Detection. During the implementation, the SMOTE technique applied on all four datasets before feeding them into the model and then trained the SGAN model.

Table 4.3 presents the results of the Baseline SGAN model after overcoming the problem of the class imbalances in the unbalanced datasets. The researcher expected to get similar results as the baseline results for the balanced datasets and significant improvements for the imbalanced datasets. However, the results were poorer than the baseline results in the both MNIST digit and MNIST fashion datasets which were somewhat expected since these two datasets are balanced. For the Pneumonia Chest X-Ray dataset, which was heavily imbalanced, the researcher got very significant improvements of almost 20% for the train data and around 7% for the test data. In the Facial Emotion Detection dataset, which was also an imbalanced dataset, the researcher did not get much improvement, which might be due to the complexity and the size of the examples in this dataset.

Table 4. 3: Results of the Baseline SGAN+SMOTE Model

Dataset	**Learning Rate**	**Batch Size**	**Train Acc.(in%)**	**Test Acc.(in %)**	**Precision**	**Recall**	**F1-score**
MNIST Digits	2e-4	128	93.32	93.313	0.93	0.93	0.93
MNIST Fashion	1e-4	256	71.40	70.72	0.69	0.69	0.69
Pneumonia Chest X-ray	2e-4	128	94.34	70.26	0.79	0.70	0.74
Facial Emotion Detection	2e-4	256	26.22	23.16	0.29	0.19	0.23

4.2.4 Results of the Optimized SGAN-SCA

Next, the researcher applied a meta-Heuristic Population Based Sine-Cosine Algorithm(SCA) for weights initialization of the discriminator and the classifier models which is the main contribution of this project. Initially the researcher planned on applying the SCA algorithm for all the models in the architectures - discriminator, classifier and generator models. But the generator model had its first layer as a Fully Connected or Dense Layer; hence it had too many trainable parameters(weights and biases). As a result, it was not possible to apply SCA to initialize the generator weights and biases. Hence, the researcher applied the SCA algorithm to the discriminator and classifier models to initialize the first layer's parameters (2D Convolutional Layer) instead of randomly initializing the weights with a SEED of 3.

Table 4.4 shows the result of the Baseline SGAN model after applying the SCA algorithm for the optimal selection of hyperparameters during the training process of the SGAN model which represents the novelty and originality of the study. As expected, the researcher got an overwhelming improvement over the normal Baseline Model results for all the datasets. In general, the Sine-Cosine algorithm is deemed one of the most potent and popular optimization algorithms in recent times.

Table 4. 4: Results of the Baseline SGAN+SCA Model

Dataset	**Learning Rate**	**Batch Size**	**Train Acc.(in %)**	**Test Acc.(in %)**	**Precision**	**Recall**	**F1-score**
MNIST Digits	2e-4	128	95.03	95.23	0.95	0.95	0.95
MNIST Fashion	1e-4	256	76.98	75.64	0.76	0.75	0.75
Pneumonia Chest X-ray	2e-4	128	93.41	86.86	0.87	0.87	0.87
Facial Emotion Detection	2e-4	256	26.64	25.32	0.3	0.22	0.25

4.2.5 Results of the Proposed Models

Finally, the researcher used the Sine-Cosine algorithm(SCA) on the data oversampled using SMOTE in the final experiment model which is named as the SGAN+SMOTE+SCA model. The researcher expected to get the best results than all the previously experimented models because the issue of the class imbalances in the unbalanced datasets had been fixed and the parameters were to be initialized using the SCA algorithm. Table 4.5 proves the significant results of the proposed model over the previous experimental **models'** results.

In more detail, the researcher got around 1% improvement in the train and test accuracy for the MNIST dataset as well as 2% improvements for the Fashion MNIST dataset in both the train and the test accuracy. In the Pneumonia Detection dataset, the researcher got an overwhelming increase of around 21% in the training accuracy and an approximately15%

increase in accuracy over the baseline results for the test accuracy. The most satisfying results were obtained in the Facial Emotion Detection Dataset where the obtained results were over 30% accuracy for both train and test data for the first time since trying out all the different combinations of methods.

Table 4. 5: Results of the Proposed Model(SGAN+SCA+SMOTE)

Dataset	**Learning Rate**	**Batch Size**	**Train Acc.(in%)**	**Test Acc.(in%)**	**Precision**	**Recall**	**F1-score**
MNIST Digits	2e-4	128	94.86	95.49	0.95	0.95	0.94
MNIST Fashion	1e-4	256	75.39	74.92	0.74	0.74	0.74
Pneumonia Chest X-ray	2e-4	128	95.48	77.69	0.81	0.78	0.80
Facial Emotion Detection	2e-4	256	31.63	30.62	0.29	0.19	0.23

4.2.6 Performance Comparison of Experiments Based on F1- Score

All the experiments' performances are summarized in the figure 4.4 where the proposed SCA algorithm showed outstanding and considerable improvements in performance over the SGAN and SGAN-SMOTE methods baselines when SCA algorithm was implemented for weight initialization. Based on the F1-score validation, the proposed model successfully showed improvements of 1%, 2%, 15%, and 5% on benchmarking multimedia datasets;(MNIST) digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection Dataset, respectively over the other SGAN Baseline models results. In more details, the proposed model performances were as expected, there has been a slight improvement when the SMOTE method was applied in the MNIST dataset. Similar observations were seen in the Pneumonia Chest X-Ray and Facial Emotion Detection datasets. However, in the Fashion-MNIST dataset, the performance of the SGAN baseline model slightly decreased after SMOTE was applied to the dataset. This might be because the dataset was already balanced, so SMOTE couldn’t make a difference to the distribution of dataset label distribution.

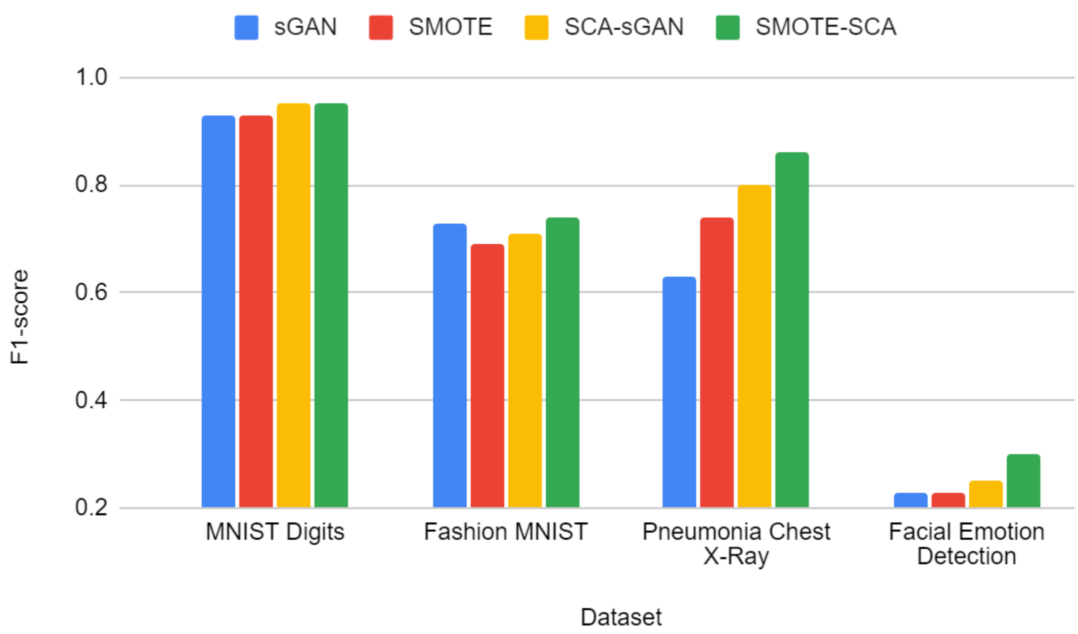

Figure 4. 4: Performance Comparison of Experiments.

4.5 Comparison of the Proposed Models with Existing Model

This section compares the proposed model with the state-of-the-art models. Different deep learning methods have been used to perform classification in the literature. Some of these methods have been used as a comparison benchmark against the final proposed model.

Table 4.6 shows the comparison of the proposed model with related literature contributions where the proposed model has considerable improvements over these different deep learning methods.For example, the proposed model gave better result more than Capsule Neural Networks (Larsen, Noever, MacVittie, & Lilly) with a test accuracy of 95.49% while the Capsule Neural Networks gave a test accuracy with 77% and 46% respectively in the MNIST dataset. Also, the proposed model gave better results than the recent developments like Multilayer Perceptron, Naïve Bayes and Decision Trees with a test accuracy of 75.39 in the MNIST fashion dataset.

In the Pneumonia Chest X-ray detection dataset, the proposed model performance is slightly better than the other models like Dark Covid, CNN ensemble and Neural Networks with 95.48% in the test accuracy. Lastly, the proposed model performance is better than the

SGD and Adam optimizers with 30% in the test accuracy of the Facial Emotion Detection dataset.

Table 4.6: Comparison of the proposed method with related literature contributions

Model		**Benchmark Dataset**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Capsule Neural Network (CapsNet) (Larsen, Noever, MacVittie, & Lilly, 2021)	N/A	MNIST Digit	0.77	0.80	0.77	0.77
Capsule Neural Network (CapsNet (contrast) (Larsen, Noever, MacVittie, & Lilly, 2021)	N/A	MNIST Digit	0.77	0.79	0.77	0.77
Graphic Convolutional Networks(GCN)(Larsen, Noever, MacVittie, & Lilly, 2021)	N/A	MNIST Digit	0.46	-	-	-
Proposed Model	Normal Baseline Results	MNIST Digit	93.48	0.93	0.93	0.93
	SMOTE Results		93.32	0.93	0.93	0.93
	SMOTE+SCA Results		94.86	0.95	0.95	0.94
Multilayer Perceptron(ML	N/A	MINST Fashion	-	0.736	0.676	0.692

P) (Zhang et al., 2018)						
Naïve Bayes (Jain & Kumar, 2020)	N/A	MINST Fashion	0.5539	0.5444	0.5539	0.5417
Decision Trees (Jain & Kumar, 2020)	N/A	MINST Fashion	0.6130	0.6085	0.6130	0.6064
Bayesian Forest (Jain & Kumar, 2020)	N/A	MINST Fashion	0.6113	0.6090	0.6113	0.5963
Proposed Model	Normal Baseline Results	MINST Fashion	74.11	0.73	0.72	0.73
	SMOTE Results		71.4	0.69	0.69	0.93
	SMOTE+SCA Results		75.39	0.74	0.74	0.74
Dark Covid Net Model (Misra et al., 2020)	N/A	Pneumonia Detection Chest X-ray	0.87.02 0.85.35	0.89.96	0.85.35	-
CNN ensemble model (Ambati & Dubey, 2021)	N/A	Pneumonia Detection Chest X-ray	91.62	-	-	-
Convolutional Neural Networks(VGG-16) (Zhang, Lipton, Li, & Smola, 2021)	Original	Pneumonia Detection Chest X-ray	0.9479	0.8512	0.9778	0.9102
	enhanced		0.9436	0.8511	0.9589	0.9018
Densely	Original	Pneumonia	0.913	0.769	0.971	0.858

Connected CNN (DenseNet121) (Zhang, Lipton, Li, & Smola, 2021)	enhanced	Detection Chest X-ray	0.934	0.804	1.00	0.891
Proposed Method	Normal Baseline Results	Pneumonia Detection from Chest X-Rays	74.77	0.81	1.00	0.63
	SMOTE Results		70.26	0.71	0.51	0.74
	SMOTE+SCA Results		95.48	0.81	0.70	0.80
SGD optimizer (Kim, Poulose, & Han, 2021)	N/A	Facial Emotion Detection Dataset	0.1298	Infinity	-	-
Adam optimizer (Kim, Poulose, & Han, 2021)	N/A	Facial Emotion Detection Dataset	20.2751	6.3535	5.2063	10.1489
Proposed Method	Normal Baseline Results	Facial Emotion Detection Dataset	25.13	0.26	0.21	0.23
	SMOTE Results		25.22	0.29	0.19	0.23
	SMOTE+SCA Results		30.62	0.29	0.19	0.23

4.6 Discussion

Based on the experiments that have been conducted, the final proposed model showed considerable improvements in the terms of classifying images with more prediction accuracy and overcoming the limitations of existing work. Apart from that, the proposed model has been used effectively to overcome the main obstacle that hardens the development of GAN in terms of the difficulty of training GAN due to being highly sensitive to the hyper parameter selections. through designing the weight optimized proposed model based on the Population-Based MetaHeuristic Sine Cosine Algorithm (SCA). Furthermore, the SCA algorithm is applied efficiently for initializing the best optimal set of the hyperparameters (weights and biases) in the proposed model' discriminator and classifier. The simple architecture of all variants of the Semi-Supervised GAN Classifier models(SGAN, SGAN+SCA, SGAN+SMOTE, and SGAN+SMOTE+SCA) can be optimized in terms of weights generation by a Sine Cosine optimization algorithm. The proposed model successfully showed improvements of 1%, 2%, 15%, and 5% on benchmarking multimedia datasets;(MNIST) digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection Dataset, respectively over the other SGAN Baseline models results.

4.7 Summary

This chapter evaluated the performance of the proposed enhanced weight-optimized Generative Adversarial Networks based on the SCA algorithm for image classification. The results presented in this chapter showed that proposed models could be successfully used for multiclass and binary classification. This proves the ability to produce a high accuracy prediction by avoiding the well-known vanishing gradient problem, which occurs during the training of simple SGANs because of multiplying the values of the randomly initialized weights with small values during the backpropagation process. The chapter then continues with the results of the enhanced model that overcomes the limitation faced by the previously proposed approaches. Hence, the proposed models generate weights that are adaptive to the model structure and dataset types. This chapter also presented the results of all the experimental models and then compared them to each other based on the train and test accuracy of the selected multimedia benchmarking datasets. The results demonstrated that the proposed model outperforms the benchmarking methods in terms of accuracy and AUC score. Lastly, the results of the proposed model have been compared with different existing deep learning methods that have been used to perform classification in the literature

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Overview

This thesis emphasizes the weights selection of recurrent neural networks based on the SCA algorithm. The main goal was to examine and enhance the performance of SGANs in analytic prediction. The proposed approaches were verified on a number of classification problems of differing sizes and contrasted with state-of-the-art approaches.

The findings indicate a consistent trend that SCA optimizers can be efficiently utilized for problems such as handwritten digits' classification, garments classification, facial emotion recognition and pneumonia detection from the chest X-Ray images. The rest of this chapter includes the contributions achieved in this thesis, emphasized together with their supported experimental results. The limitations of this research are then discussed and nominated for potential future research directions.

5.2 Research Summary

In this work, the researcher proposed an SGAN classifier model that performs binary as well as multiclass classification accurately with the help of the Population-Based Meta-Heuristic Sine Cosine Algorithm (SCA). Previous works majorly focused on a single dataset that implemented the proposed architecture on four different benchmark datasets to show the efficiency of the proposed model. The multimedia benchmarking datasets used were the MNIST digits, the MNIST Fashion, Pneumonia Detection from Chest X-Rays and Facial Emotion Detection with a set of existing evaluation classification metrics for validating and illustrating the efficiency of the proposed model.

5.3 Achieved Objectives

The main objective of this study was to figure out the effects of weights and bias initialization in semi-supervised generative adversarial network classifiers and overcome the limitation of SGAN in selecting hyperparameters settings during the training process of the SGAN model with the use of Sine Cosine Algorithm (SCA) for the initialization of the parameters of the first layer (the weights and the biases). Therefore, all experimental models showed that the proposed technique could be effectively used in the terms of classifying images with more prediction accuracy and overcoming the limitations of existing work. The final proposed model has shown outstanding performance in terms of accuracy and AUC score and overwhelmingly better accuracy over the other three models. Moreover, the proposed model successfully showed improvements of 1%, 2%, 15%, and 5% on benchmarking multimedia datasets;(MNIST) digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection Dataset, respectively over the other SGAN Baseline models results.

Firstly, after achieving the task of survey and analysis of the state-of-the-art Generative Adversarial Networks classification models, the researcher has designed the Original GAN model through a typical and full GAN model using a python environment and then trained the GAN model in MNIST dataset. The GAN model' improvement in generation of the images over iterations was very strong with 95% in accuracy, precision, recall and F1-score respectively.

Secondly, the researcher designed the SGAN classifier model as an extension of the Generative Adversarial Network architecture for addressing semi-supervised learning problems. Therefore, the Baseline results consist of the classification results on the four datasets whereby the model was a simple Semi-Supervised Generative Adversarial Network classifier that was directly trained and tested on the pixel value tables of the dataset's images. The results of the Baseline SGAN model show that benchmark datasets like the MNIST digits dataset gave superior results under Baseline conditions, majorly due to how easy it was to classify handwritten digits, given how complex and fine-tuned the SGAN classifier is. However, when the difficult datasets are taken like the Facial Emotion Detection dataset wherein the predicted facial emotion of a person from the image of the face provided, the same complex SGAN classifier became counterproductive and led to inaccurate results in most cases. As it is known that GANs are difficult to train, and it's difficult to achieve good results at times.

After that, it found that the datasets Pneumonia Chest X-Ray and Facial Emotion Detection had class imbalances leading to poorer results than expected. So, the researcher used the Synthetic Minority Oversampling Technique (SMOTE) for overcoming this problem. Hence, the researcher applied the SMOTE technique with the SGAN classifier model. However, the results were poorer than the baseline results in the both MNIST digit and MNIST fashion datasets which were somewhat expected since these two datasets are balanced. Aside from that, the researcher got very significant improvements of almost 20% for the train data and around 7% for the test data for the Pneumonia Chest X-Ray dataset. In the Facial Emotion Detection dataset, the researcher did not get much improvement, which might be due to the complexity and the size of the examples in this dataset.

Next, the SGAN+ SCA model was developed through applied a meta-Heuristic Population Based Sine-Cosine Algorithm(SCA) for weights initialization of the discriminator and the classifier models. As expected, the researcher got an overwhelming improvement over the normal Baseline Model results for all the datasets. In general, the Sine-Cosine algorithm (SCA) is deemed one of the most potent and popular optimization algorithms in recent times.

Lastly, the researcher used the Sine-Cosine algorithm(SCA) on the data oversampled using SMOTE in the final experiment model which was named as the SGAN+SMOTE+SCA model. The researcher got the best results than all the previously experimented models because the issue of the class imbalances in the unbalanced datasets had been fixed and the parameters were to be initialized using the SCA algorithm. The Results of the proposed model were very significant compared to the previous experimental models results. In more detail, the researcher got around 1% improvement in the train and test accuracy for the MNIST dataset as well as 2% improvements for the Fashion MNIST dataset in both the train and the test accuracy. In the Pneumonia Detection dataset, the researcher got an overwhelming increase of around 21% in the training accuracy and an approximately15% increase in accuracy over the baseline results for the test accuracy. The most satisfying results were obtained in the Facial Emotion Detection Dataset where the obtained results were over 30% accuracy for both train and test data for the first time since trying out all the different combinations of methods.

5.4 Summary of Obtained Results

The main contribution of this study was to enhance the hyperparameter of semi-supervised GAN classifiers-based Sine Cosine algorithm for multimedia datasets. Therefore, the architectures were used for binary and multiclass image classification on four benchmark datasets.

The summary of the obtained results of this research were extensive, exhaustive literature and code reviews have been done to use different weight optimization techniques for the SGAN classifiers for identifying the research gaps that existed in the literature. The final weight-optimized SCAN classifier was enhanced based on Sine-Cosine Algorithm optimization in the Multimedia datasets for the optimal selection of hyperparameters during the training process of the final model which represented the novelty and originality of the study. Aside from that, the Grid Search (GS) was efficiently applied for each experimental setup model to find the best optimal set of hyperparameters (Learning rate & Batch size) by locating and picking the most appropriate values for the dataset size and model structure where the researcher got the most optimal results. SMOTE technique used with all experimental models for addressing the class imbalances in the unbalanced datasets which further led to a substantial improvement in the proposed model's performance. However, PCA technique did not yield better results. One of the reasons why PCA might have failed was that the differentiating characteristics of the classes were not reflected in the variance of the variables. This was because PCA did not take into account class information when calculating the principal components. Furthermore, the four well-known multimedia benchmark datasets ((MNIST digits, Fashion MNIST, Pneumonia Chest X-ray, and Facial Emotion Detection) and a set of existing evaluation classification metrics were used for validating and showing the efficiency of the proposed model. Besides that, the proposed model successfully showed improvement in the test accuracy scores on all the benchmarking multimedia datasets respectively. Finally, the proposed model was compared against existing models, the results on each dataset were recorded and demonstrated the effectiveness of the proposed model in the classification accuracy in many cases.

5.5 Future Works

This work could be further expanded by implementing the SCA algorithm for the initialization of weights and biases of the Generator efficiently without making it so computationally expensive. This work could be experimented with more benchmark datasets such as CIFAR-10 and CIFAR-100. Moreover, Auto-Encoders technique could be used where the sheer number of parameters would benefit from their effective initialization through SCA. Finally, Further Metaheuristic Optimization algorithms (e.g Bat algorithm) can be used to improve the SGAN model for the problem of hyperparameter selections.

REFERENCES

Abdi, A., Shamsuddin, S. M., Hasan, S., & Piran, J. (2018). Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment. Expert Systems with Applications, 109, 66-85.

R. Agarwal., The 5 Classification Evaluation metrics every Data Scientist must know", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226. [Accessed: 03- Feb-2022].

Albaom Ma, Sidi11 Fa, Jabar Ma, Abdullah R, Ishak I, Yunikawati Na, Purboyo M, Priambodo Jh, Issac O, Al-Harasi Ah. (2021). The Impact of Tourist's Intention to Use Web 3.0: A Conceptual Integrated Model Based on TAM & DMISM. Journal of Theoretical and Applied Information Technology, 99(24).

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164* ,9(3).

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164, 3(3).

Al-Tashi, Q., Abdulkadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2020). Approaches to multi-objective feature selection: A systematic literature review. IEEE Access, 8, 125076-125096.

Al-Tashi, Q., Abdulkadir, S. J., Rais, H. M., Mirjalili, S., Alhussian, H., Ragab, M. G., & Alqushaibi, A. (2020). Binary multi-objective grey wolf optimizer for feature selection in classification. IEEE Access, 8, 106247-106263.

Al-Tashi, Q., Rais, H., & Jadid, S. (2018). Feature selection method based on grey wolf optimization for coronary artery disease classification. In International conference of reliable information and communication technology (pp. 257-266). Springer.

Ambati, A., & Dubey, S. R. (2021). Ac-covidnet: Attention guided contrastive CNN for recognition of covid-19 in chest x-ray images. arXiv preprint arXiv:2105.10239.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

Barlow, H. B. (1989). Unsupervised learning. Neural computation, 1(3), 295-311.

Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems, 28: 1171–1179

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In Neural networks: Tricks of the trade (pp. 437-478). Springer.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. Computational Statistics & Data Analysis, 120: 70-83.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

Borgelt, C. (2003). Efficient implementations of apriori and eclat. In FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations (p. 90).

Bottou, L. (2012). Stochastic gradient descent tricks. In Neural networks: Tricks of the trade (pp. 421-436). Springer.

Bozkurt, E. (2021). *Machine Learning Classification Algorithms with Codes*. Medium. Retrieved 14 March 2022, from https://medium.com/analytics-vidhya/machine-learning classification-algorithms-with-codes-5a8af4491fcb.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Brownlee, J. (2021). *Weight Initialization for Deep Learning Neural Networks*. Machine Learning Mastery. Retrieved 3 January 2022, from https://machinelearningmastery.com/weight-initialization-for-deep-learning-neural-networks/.

Cerqueira, V., Torgo, L., Smailović, J., & Mozetič, I. (2017). A comparative study of performance estimation methods for time series forecasting. In 2017 IEEE international conference on data science and advanced analytics (DSAA) (pp. 529-538). IEEE.

Chatterjee, S., & Hadi, A. S. (2013). Regression analysis by example: John Wiley & Sons.

Chavdarova, T., & Fleuret, F. (2018). Sgan: An alternative training of generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9407-9415).

Chen, S. A., Li, C. L., & Lin, H. T. (2021). A Unified View of cGANs with and without Classifiers. *Advances in Neural Information Processing Systems, 34*.

Cherkassky, V., & Ma, Y. (2002). Selection of meta-parameters for support vector regression. In International Conference on Artificial Neural Networks (pp. 687-693). Springer.

Chiroma, H., Abdullahi, U. A., Alarood, A. A., Gabralla, L. A., Rana, N., Shuib, L., ... & Herawan, T. (2018). Progress on artificial neural networks for big data analytics: a survey. IEEE Access, 7: 70535-70551.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine, 29(6), 141-142.

Dietterich, T. G. (2002). Machine learning for sequential data: A review. In Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), pages 15–30. Springer.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

Dozat, T. (2016). Incorporating nesterov momentum into adam. 12 (2).

Elkhalil, K., Kammoun, A., Couillet, R., Al-Naffouri, T. Y., & Alouini, M. S. (2020). A large dimensional study of regularized discriminant analysis. *IEEE Transactions on Signal Processing*, *68*: 2464-2479.

Feng, G. H. (2011). Support Vector Machine parameter selection method. Computer Engineering and Applications, 47(3), 123.

Flores, S. (2019). *Variational Autoencoders are Beautiful | Blogs*. Compthree.com. Retrieved 15 April 2022, from https://www.compthree.com/blog/autoencoder/.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811.

Gong, M., Xu, Y., Li, C., Zhang, K., & Batmanghelich, K. (2019). Twin auxilary classifiers gan. *Advances in neural information processing systems*, *32*: 4(11).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems,27: 10(6).

Hadi, A. S., & Chatterjee, S. (2015). Regression analysis by example. John Wiley & Sons. 158-384.

Haque, A. (2020). Ec-gan: Low-sample classification using semi-supervised algorithms and gans. arXiv preprint arXiv:2012.15864.

Haque, A. (2020). Ec-gan: Low-sample classification using semi-supervised algorithms and gans. arXiv preprint arXiv:2012.15864.

Hinz, T., Navarro-Guerrero, N., Magg, S., & Wermter, S. (2018). Speeding up the hyperparameter optimization of deep convolutional neural networks. International Journal of Computational Intelligence and Applications, 17(02), 1850008.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

HOLMGREN, A., & MORGAN, F. J. (1976). Enzymic reduction of disulfide bonds by thioredoxin: The reactivity of disulfide bonds in human choriogonadotropin and its subunits. European journal of biochemistry, 70(2), 377-383.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. 15(4).

Hui, J. (2018). GAN A comprehensive review into the gangsters of GANs (Part 1). Medium. Retrieved 10 May 2020, from https://jonathan-hui.medium.com/gan-a-comprehensive-review-into-the-gangsters-of-gans-part-1-95ff52455672.

Hui, J. (2018). GAN Why it is so hard to train Generative Adversarial Networks, Medium. Retrieved 14 May 2019, from https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b.

Hwang, U., Jung, D., & Yoon, S. (2019). Hexagan: Generative adversarial nets for real world classification. In International Conference on Machine Learning (pp. 2921-2930). PMLR.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. Data mining and knowledge discovery, 33(4), 917-963.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

Jain, S., & Kumar, V. (2020). Garment categorization using data mining techniques. Symmetry, 12(6), 984.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4401-4410).

Khanmohammadi, S., Adibeig, N., & Shanehbandy, S. (2017). An improved overlapping k-means clustering method for medical applications. Expert Systems with Applications, 67: 12-18.

Kim, J. H., Poulose, A., & Han, D. S. (2021). The extensive usage of the facial image threshing machine for facial emotion recognition performance. Sensors, 21(6), 2026.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. In Automated Machine Learning (pp. 81-95). Springer.

Kumar, S. K. (2017). On weight initialization in deep neural networks. arXiv preprint arXiv:1704.08863.

Längkvist, M., Karlsson, L., & Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters, 42, 11-24.

Larsen, E., Noever, D., MacVittie, K., & Lilly, J. (2021). Overhead-MNIST: Machine Learning Baselines for Image Classification. arXiv preprint arXiv:2107.00436.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5457-5466).

Li, Y. F., & Liang, D. M. (2019). Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, *13*(4), 669-676.

Li, Y., Xiao, N., & Ouyang, W. (2018). Improved boundary equilibrium generative adversarial networks. IEEE access, 6, 11342-11348.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2018). Are gans created equal? a large-scale study. Advances in neural information processing systems, 31: 698-07

Ma, X. X., & Wang, J. S. (2018). Optimized parameter settings of binary bat algorithm for solving function optimization problems. Journal of Electrical and Computer Engineering, 14(5). 1-9.

Machinery, C. (1950). Computing machinery and intelligence-AM Turing. Mind, 59: 236-433.

Mahapatra, D., Bozorgtabar, B., & Garnavi, R. (2019). Image super-resolution using progressive generative adversarial networks for medical image analysis. Computerized Medical Imaging and Graphics, 71, 30-39.

Maier, H. R., & Dandy, G. C. (1998). The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study. Environmental Modelling & Software, 13(2), 193-209.

Martens, J., & Sutskever, I. (2012). Training deep and recurrent networks with hessian-free optimization. In Neural networks: Tricks of the trade (pp. 479-535). Springer.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-based systems, 96, 120-133.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.

Mirjalili, S., Song Dong, J., Lewis, A., & Sadiq, A. S. (2020). Particle swarm optimization: theory, literature review, and application in airfoil design. Nature-inspired optimizers, 167-184.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Misra, S., Jeon, S., Lee, S., Managuli, R., Jang, I. S., & Kim, C. (2020). Multi-channel transfer learning of chest X-ray images for screening of COVID-19. Electronics, 9 (9), 1388.

Moisen, G. G. (2008). Classification and regression trees. In: Jørgensen, Sven Erik; Fath, Brian D.(Editor-in-Chief). Encyclopedia of Ecology, volume 1. Oxford, UK: Elsevier. p. 582-588.

Mooney, P. (2018). Chest X-Ray Images (Pneumonia). Mooney. Retrieved 7 September 2021, from https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

Nawi, N. M., Khan, A., Rehman, M. Z., Chiroma, H., & Herawan, T. (2015). Weight optimization in recurrent neural networks with hybrid metaheuristic Cuckoo search techniques for data classification. Mathematical Problems in Engineering. 10(5), 1-12.

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583.

Odena, A., Olah, C., & Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In International conference on machine learning (pp. 2642-2651). PMLR.

Petneházi, G. (2019). Recurrent neural networks for time series forecasting. arXiv preprint arXiv:1901.00069.

Petzold, C. (2008). The annotated Turing: a guided tour through Alan Turing's historic paper on computability and the Turing machine: Wiley Publishing. 22: 327-383.

Plakias, S., & Boutalis, Y. S. (2019). Exploiting the generative adversarial framework for one-class multi-dimensional fault detection. Neurocomputing, 332, 396-405.

Ragab, M. G., Abdulkadir, S. J., & Aziz, N. (2020). Random search one dimensional CNN for human activity recognition. In 2020 International Conference on Computational Intelligence (ICCI) (pp. 86-91). IEEE.

Ragab, M. G., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., Alyousifi, Y., Alhussian, H., & Alqushaibi, A. (2020). A novel one-dimensional cnn with exponential adaptive gradients for air pollution index prediction. Sustainability, 12(23), 10090.

Rashidi, H. H., Tran, N., Albahra, S., & Dang, L. T. (2021). Machine learning in health care and laboratory medicine: General overview of supervised learning and Auto-ML. *International Journal of Laboratory Hematology*, *43*, 15-22.

Reddy, Y. C. A. P., Viswanath, P., & Reddy, B. E. (2018). Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, *7*(1.8), 81.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. (1986). Sequential thought processes in PDP models. Parallel distributed processing: explorations in the microstructures of cognition, 2: 3-57.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29.

Sambare, M. (2021). FER-2013. Kaggle.com. Retrieved 16 November 2021, from https://www.kaggle.com/msambare/fer2013.

Schreiber, J., Jessulat, M., & Sick, B. (2019). Generative adversarial networks for operational scenario planning of renewable energy farms: A study on wind and photovoltaic. In International Conference on Artificial Neural Networks (pp. 550-564). Springer.

Shekar, B. H., & Dagnew, G. (2019, February). Grid search-based hyperparameter tuning and classification of microarray cancer data. In 2019 second international conference on advanced computational and communication paradigms (ICACCP) (pp. 1-8). IEEE.

Shuai, Y., Zheng, Y., & Huang, H. (2018). Hybrid software obsolescence evaluation model based on PCA-SVM-GridSearchCV. In 2018 IEEE 9th international conference on software engineering and service science (ICSESS) (pp. 449-453). IEEE.

Somu, N., MR, G. R., & Ramamritham, K. (2020). A hybrid model for building energy consumption forecasting using long short term memory networks. Applied Energy, 261, 114131.

Sui, B., Jiang, T., Zhang, Z., & Pan, X. (2020). ECGAN: An improved conditional generative adversarial network with edge detection to augment limited training data for the classification of remote sensing images with high spatial resolution. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 1311-1325.

Sun, W., Dai, G., Zhang, X., He, X., & Chen, X. (2021). TBE-Net: A Three-Branch Embedding Network With Part-Aware Ability and Feature Complementary Learning for Vehicle Re-Identification. IEEE Transactions on Intelligent Transportation Systems. 3(12), 1-13.

Sun, W., Dai, L., Zhang, X., Chang, P., & He, X. (2021). RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. Applied Intelligence, 1-16.

Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the turing test (pp. 23-65). Springer.

Vincent, J. (2019). This Person Does Not Exist. com uses AI to generate endless fake faces. The Verge, 15: 1-15.

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3), 279-292.

Wei, J. (2019). Forget the learning rate, decay loss. arXiv preprint arXiv:1905.00094.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), 1550-1560.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.

Yao, Z., Zhang, B., Wang, Z., Ouyang, W., Xu, D., & Feng, D. (2019). IntersectGAN: Learning Domain Intersection for Generating Images with Multiple Attributes. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 1842-1850).

Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. Medical image analysis, 58, 101552.

Yu, C., Li, Y., Bao, Y., Tang, H., & Zhai, G. (2018). A novel framework for wind speed prediction based on recurrent neural networks and support vector machine. Energy Conversion and Management, 178: 137-145.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., & Atkinson, P. M. (2018). A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS Journal of Photogrammetry and Remote Sensing, 140: 133-144.

Zhao, D., Guo, B., & Yan, Y. (2019). Parallel image completion with edge and color map. Applied Sciences, 9(18), 3856.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130.

LIST OF PUBLICATIONS

The research work in this thesis has been presented in formal proceedings and published in the following articles:

Journal Publications

1. **Al-Ragehi. A** [1], Kadir, S. J. A [1,2]*, A Muneer [1,2], Sadeq, S [3], Al-Tashi[4,5] (2022). Hyper-Parameter Optimization of Semi-Supervised GANs Based-Sine Cosine Algorithm for Multimedia Datasets. Comput. Mater. Contin, 73, 2.
   [Q2, IF 3.772], [ISI, Scopus]