

**DEVELOPING A HIGH-PERFORMANCE SOIL FERTILITY STATUS
PREDICTION VOTING ENSEMBLE USING BRUTE EXHAUSTIVE
OPTIMIZATION IN AUTOMATED MULTIPRECISION WEIGHTS OF
HYBRID CLASSIFIERS**

Augustine Josephat Malamsha

**A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Information and Communication Science and Engineering of
the Nelson Mandela African Institution of Science and Technology**

Arusha, Tanzania

August, 2023

ABSTRACT

With the advent of machine learning (ML) techniques, various algorithms have been applied in previous studies to develop models for predicting soil fertility status. However, these models are observed to use varying fertility target classes, and variations have been reported in these models' predictive performances. As a result, practical applications of these models for obtaining the most accurate predictions may become hindered. While the weighted voting ensemble (WVE) ML technique can be used to improve soil fertility status prediction by aggregating individual models prediction, guaranteeing finding of an optimal WVE assignment weights is challenging. Whereas a brute exhaustive search procedure can be applied for the mentioned task, there is a lack of exploration on the exploitation of automated classifiers' precise weights combinations as search spaces for successful optimization. This research aims to develop a high-performance soil fertility status prediction voting ensemble using brute exhaustive optimization in automated 1EXP(-)Z⁺ multi-precision weights of hybrid classifiers. Soil chemical properties and ML modeling algorithms for modeling soil fertility status were identified. Base hybrid ML classification models for predicting soil fertility status were evaluated using Tanzania as a case study. Finally, the base ML hybrids WVE models were optimized using brute exhaustive search procedure's novel developed search spaces generation algorithm for guaranteed optimal solution finding. The research was designed using design science research methodology, with the application of unsupervised machine learning K-mean algorithm with a knee detection method to find the optimal number of soil fertility status target classes, and supervised learning algorithms were applied to model classifiers for those optimal classes. Three soil fertility target classes were identified by clustering technique. The model achieved on test data a predictive accuracy of 98.93%, with respective AUC of 82%, 83%, and 87% for low, medium, and high soil fertility targets classes. Whereas these performances are observed higher compared to models in previous studies, 92% correct classifications were obtained on validation against external unseen laboratory-based tested soil results. Therefore, soil testing laboratories and farmers should consider using the model to smartly manage soil fertility which may lead to improved crop growth and productivity. The government could set agricultural-related policies that require the use of the model by farmers with the provision of agricultural inputs subsidies. Future work could be to develop an integrated real-time web and mobile application for providing farmers with soil fertility status information.

DECLARATION

I, **Augustine Josephat Malamsha**, do hereby declare to the Senate of the Nelson Mandela African Institution of Science and Technology that this thesis is my original work and that it has neither been submitted nor being concurrently submitted for a degree award in any other institution.

Augustine Josephat Malamsha



27.08.2023

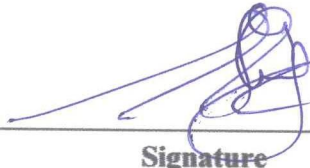
Name of Candidate

Signature

Date

The above declaration is confirmed by:

Dr. Mussa Ally Dida



31/08/2023

Name of Supervisor 1

Signature

Date

Prof. Sabine Moebs



27/08/2023

Name of Supervisor 2

Signature

Date

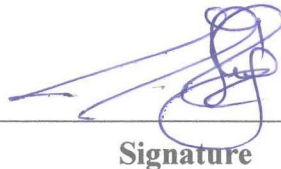
COPYRIGHT

This dissertation is copyright material protected under the Berne Convention, the Copyright Act of 1999, and other international and national enactments, on that behalf, of intellectual property. It may not be reproduced by any means, in full or in part, except for short extracts in fair dealings; for research or private study, critical scholarly review or discourse with an acknowledgment, without written permission of the Deputy Vice Chancellor for Academic, Research, and Innovation, on behalf of both the author and the Nelson Mandela African Institution of Science and Technology.

CERTIFICATION

The undersigned certify that they have read and hereby recommend for acceptance by the Nelson Mandela African Institution of Science and Technology a dissertation entitled: *Developing a High-Performance Soil Fertility Status Prediction Voting Ensemble using Brute Exhaustive Optimization in Automated MultiPrecision Weights of Hybrid Classifiers*, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology.

Dr. Mussa Ally Dida



31/08/2023

Name of Supervisor 1

Signature

Date

Prof. Sabine Moebs



31/08/2023

Name of Supervisor 2

Signature

Date

ACKNOWLEDGMENTS

I acknowledge the assistance received from many people and appreciate all those who assisted me during my studies at the Nelson Mandela African Institution of Science and Technology (NM-AIST). I would commence by expressing my uttermost gratitude to almighty God, for sustaining my life, strengthening, and blessings me to completion of this dissertation.

Next would like to thank my supervisors, Dr. Mussa Ally Dida and Prof. Sabine Moebs, they enabled me to develop the right mindset and acquire the relevant skills and experience needed to carry out quality research, which I did not have before the commencement of my PhD studies, which turn that gave me the autonomy to manage this research, enabling me to scrutinize research problems while they continuously offered valuable guidance whenever I encountered challenges.

My thanks also go out to other faculty and supporting staff, as well as other colleagues at the NM-AIST. I wish to thank the Dean of the School of life sciences Prof. Kelvin Mark Mtei, Prof. Shubi Kaijage, Dr. Bonny Mgawe, Dr. Devotha Nyambo, and Dr. Judith Leo from the School of Computational and communication sciences and Engineering, for their valuable inputs.

In addition, I would like to thank Mr. Adam Mawenya (NM-AIST laboratory engineer) for the computational hardware setup, and also Dr. Michael Mollel, as well as my study colleagues at NM-AIST cohort 7 for their valuable suggestions and motivation which influenced this work. Great gratitude also goes to other institution's key personnel including Dr. Jaha Mvula of the electronic governance agency (eGA) of Tanzania, Dr. Meliyo of the Tanzania Agricultural Research Institute (TARI) Hombolo, as well as the director of TARI SELIAN.

I also highly thank the soil analyst of the Soil Care Depart of the Live Support Systems (T) LTD (LSSL) Soil Services Company, Mr. Antony Prosper, for the provision of access to Njombe's soil laboratory data test results that were used for validation in this research.

I would further like to express my profound gratefulness to African Development Bank (AfDB) for sponsoring my studies, and to the Institute of Finance Management (IFM) which this research would not have completed without it. Sincere gratitude goes to the NM-AIST AfDB project manager, Mr. Julius Lenguyana for his spontaneous outpouring of support and cooperation.

Special thanks to my parents Josephat Malamsha, and Eda Mwangoka Malamsha, who ensured that I had an outstanding educational achievement, being down to earth I am grateful for having you in my life.

Finally, I appreciate all those who made an outreach toward the accomplishment of this study.

DEDICATION

This dissertation work is dedicated to my lovely family, the Josephat Malamsha's, who have been supporting and encouraging me during graduate school and general life challenges.

TABLE OF CONTENTS

ABSTRACT.....	i
DECLARATION	ii
COPYRIGHT.....	iii
CERTIFICATION	iv
ACKNOWLEDGMENTS	v
DEDICATION.....	vii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS AND SYMBOLS	xviii
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background of the Problem	1
1.2 Statement of the Problem.....	3
1.3 Rationale of the Study.....	5
1.4 Research Objectives.....	7
1.4.1 General Objective	7
1.4.2 Specific Objectives	7
1.5 Research Questions	8
1.6 Significance of the Study	9
1.7 Delineation of the Study	10
CHAPTER TWO	12
LITERATURE REVIEW	12
2.1 A Theoretical Literature Review.....	12
2.1.1 Agricultural Soils Properties for Fertility Status Analysis.....	12
2.1.2 Machine Learning Overview	13

2.1.3	Hybrid Machine Learning.....	30
2.1.4	Ensemble Learning	32
2.1.5	Bagging	33
2.1.6	Boosting	35
2.1.7	Weighted Voting Ensemble Scheme for Model Performance Improvement ..	36
2.1.8	Brute Exhaustive Search Algorithm	39
2.2	Empirical Literature Review	41
2.2.1	Application of Machine Learning Techniques in Modelling Agricultural Soil Nutrients and Other Chemical Properties for Fertility Status Prediction	41
2.2.2	Summary of the Empirical Review	45
2.2.3	Research Gap	48
CHAPTER THREE		50
MATERIALS AND METHODS.....		50
3.1	Introduction.....	50
3.2	Design Science Research	50
3.2.1	Problem Identification and Motivation	52
3.2.2	Define the Objectives of a Solution	52
3.2.3	Design and Development	53
3.2.4	Demonstration.....	53
3.2.5	Evaluation	54
3.2.6	Communication.....	54
3.3	Materials and Methods.....	54
3.3.1	Study Area	54
3.3.2	Modeling Dataset	55
3.3.3	Heterogeneous Hybrid’s Weighted Voting Ensemble Experiment Setups.....	64
3.3.4	Base Models Performance Improvement Through WVE.....	68

3.3.5	Development of the High Performance put Brute Exhaustive WVE 1EXP (-) Z+ Optimization Algorithm.....	69
3.3.6	Development Environment	79
3.3.7	Mathematical Co-processor Computational Limitations	79
3.3.8	Performance Evaluation.....	81
3.3.9	Validation and Utility of the Model.....	85
CHAPTER FOUR.....		87
RESULTS AND DISCUSSION		87
4.1	Data Pre-processing Results	87
4.1.1	Descriptive Statistics of Used Dataset	87
4.1.2	Quality of Dataset	89
4.2	Modeling and Optimization Results	91
4.2.1	Soil Fertility Index derivation.....	91
4.2.2	Base Models: Performance Evaluation	96
4.2.3	Computational Complexity: Efficiencies.....	98
4.2.4	EXP (-) Z ⁺ _{IT} -ASMPSS-BEO- _{IS} WVE Optimization Results.....	100
4.2.5	The 1EXP (-) Z ⁺ _{IT} -ASMPSS-BEO- _{IS} WVE Effectiveness	104
4.2.6	Comparative Analysis: The WVE vs Individual Models Results.....	112
4.3	The WVE Model Validation and Utility Evaluation	117
4.3.1	Model Validation Results	118
4.3.2	Model Utility: WVE Model Predictions for Maize Plantations Field Grain Yields Experimentation Results.....	121
4.4	Discussion	124
CHAPTER FIVE		129
CONCLUSION AND RECOMMENDATIONS		129
5.1	Conclusion	129
5.2	Recommendations.....	131

REFERENCES	134
APPENDICES	155
RESEARCH OUTPUTS.....	162

LIST OF TABLES

Table 1:	Summary of ML algorithms strengths and weaknesses	29
Table 2:	Genetic, greedy and brute experimental computational times.....	40
Table 3:	Summary of the State-of-the-art ML-based approaches and Soil chemical properties used in modeling nutrients and fertility status prediction.....	45
Table 4:	Description of the agricultural soil properties and maize yields dataset features..	55
Table 5:	Tanzania Agricultural Research Institute and Tanzania Ministry of Agriculture’s respective Agricultural Soils Raw Data with corresponding maize grain yields..	57
Table 6:	Soil care laboratory-based validation dataset	62
Table 7:	Model utility evaluation field experimentation soil properties collected data	63
Table 8:	The four basic confusion matrix metrics	82
Table 9:	Some supervised learning performance metrics	83
Table 10:	Experimental plantation plan.....	86
Table 11:	Statistical description of the used agricultural soils and yield data	88
Table 12:	The HHCM performance	97
Table 13:	Search space precisions, formulation, and containment and optimization times	101
Table 14:	Results of the effectiveness of the proposed search heuristic procedure.....	106
Table 15:	Total harvest tons amounts of maize per quarter by study section.....	123

LIST OF FIGURES

Figure 1: Global population size: estimates, 1950-2022	1
Figure 2: Soil Assessment Methods Evolution.....	2
Figure 3: State of art search heuristics theoretical guaranteeing and non-guaranteeing of optimality tapping	6
Figure 4: Depiction of the study rationale	11
Figure 5: The AI Agent System Architecture	14
Figure 6: Non-Exhaustive Taxonomy of the different main ML algorithms.....	17
Figure 7: Traditional ML Workflow	18
Figure 8: Input-output learning ML function.....	18
Figure 9: Graphical Network Graph Model Naive Bayes' classifier assuming independent input attributes	19
Figure 10: An 8-nearest neighbors decision problem.....	21
Figure 11: A decision tree.....	22
Figure 12: Support Vector Machine Maximum Margin	23
Figure 13: Artificial neural network layers with sigmoid function	25
Figure 14: K-Mean clustering	28
Figure 15: The Conceptual framework of a data manipulation HML workflow.....	32
Figure 16: Random forest classifier.....	34
Figure 17: The Architecture of Gradient Boosting	35
Figure 18: Gradient boosting's descent explanatory	36
Figure 19: Schematic overview of the weighting algorithm.....	37
Figure 20: Previous studies' results of brute exhausted search exhibiting superiority just like genetics and greed.....	39
Figure 21: Soil parameters use frequency.....	48
Figure 22: Algorithms use frequency	48
Figure 23: The DSR methodology cycles.....	51

Figure 24: The ML and optimization-related reviewed papers publications	53
Figure 25: Study experimentation location.....	54
Figure 26: Laboratory-based off-the-shelf soil fertility test results – low	59
Figure 27: Laboratory-based off-the-shelf soil fertility test results – adequate.....	60
Figure 28: Laboratory-based off-the-shelf soil fertility test results – high.....	61
Figure 29: The 2S-HHEC Experimental setup	65
Figure 30: The proposed multi-precision weights formulation operational architecture	72
Figure 31: The full 1EXP (-) Z ⁺ IT-ASMPSSA_BES_ISWVEO flowchart	76
Figure 32: The 1EXP (-) Z ⁺ IT -ASMPSS-BEO-ISWVE package diagram	78
Figure 33: The 1EXP (-) Z ⁺ IT-ASMPSS-BEO-ISWVE experimental setup	79
Figure 34: Micro-processor FPU	80
Figure 35: Quantum’s qubit continuum state vs classical bit information representation.....	81
Figure 36: Comparison of observations to features ratio of the used dataset in similar ML implementations.....	89
Figure 37: pH QQ Plot.....	90
Figure 38: S QQ plot.....	90
Figure 39: OC QQ plot	91
Figure 40: P QQ plot.....	91
Figure 41: Knee elbow = 3.....	92
Figure 42: Test Results for the Analysis of Variance between different fertility groups	92
Figure 43: The ANOVA (Tukey Test) results	93
Figure 44: Organic carbon, pH vs Fertility index clusters visualization	93
Figure 45: Organic carbon, Calcium vs Fertility index Clusters visualization.....	94
Figure 46: Organic carbon, Phosphorus vs Fertility index Clusters visualization.....	94
Figure 47: Organic carbon, electrical conductivity vs fertility index clusters visualization ..	95
Figure 48: Soil Fertility classes (label) distribution.....	95

Figure 49: Correlation heatmap of the soil chemical properties	96
Figure 50: The 1EXP (-) Z^+ based Sequence Initial term function asymptotic optimality to WVE weights constraints	98
Figure 51: Hardware clock time in search space precision 1 and 2	102
Figure 52: Hardware clock time limitation in search space 3	102
Figure 53: 1EXP (-) Z^+ IT-ASMPSS-BEO-ISWVE sequences formulations and optimization Algorithm Efficiency in the most stable search space reference $Z^+ = 2$	103
Figure 54: Total optimization time in search spaces	103
Figure 55: The WVE initial and filtered potential search combinations in the stable domain search space 1 and 2	105
Figure 56: Search spaces precision effect on WVEs accuracies	108
Figure 57: Best WVE accuracies in Space $Z^+ 1$ and $Z^+ 2$	109
Figure 58: The DT and KNN combination ROC plots and AUC scores	110
Figure 59: The RF, SV, and KNN combination ROC plots and AUC scores	110
Figure 60: The GB, RF, SVM, and KNN combination ROC plots and AUC scores	111
Figure 61: The GB, DT, RF, SVM, and KNN combination ROC plots and AUC scores	111
Figure 62: The 2S-HHEC's (WVE) and learners' performances	112
Figure 63: Visual display of the model performances as compared to benchmark 1	114
Figure 64: The ROC curves for the proposed model as compared to benchmark 2	115
Figure 65: High fertility class prediction ROC curves for the proposed model as compared to benchmark 2	116
Figure 66: Medium Fertility Class Prediction ROC curves for the proposed model as compared to benchmark 2	116
Figure 67: Low Fertility Class Prediction ROC curves for the proposed model as compared to benchmark 2	117
Figure 68: Streamlit-based interface for batch uploading soil properties data file into the model for prediction	118
Figure 69: Plot of the WVE predictions vs Actual Soil Laboratory Test results	119

Figure 70: Summary of the soil fertility statuses WVE model predictions on Soil laboratory validation by numerical percentages	120
Figure 71: Plot of the Summary of soil fertility statuses predictions by WVE model using Soil laboratory validation dataset.....	120
Figure 72: Percentage of soil fertility status constituents in the 64 model-based predictions section samples	121
Figure 73: Plot of the percentages of predicted samples	122
Figure 74: Percentage-wise proportions of the total harvest in each study section	123
Figure 75: Harvest in each study section, totals, and extrapolation on an acre	124
Figure 76: Comparison of the extrapolated 1-acre maize harvest value with the Tanzania 2019-2020 Agricultural year in tons per acre, in Tanzania Mainland	127

LIST OF APPENDICES

Appendix 1:	Research Validation Data Provision Letter.....	155
Appendix 2:	Python code for Fertility Index Derivation.....	156
Appendix 3:	Python code for Base Models Evaluation.....	157
Appendix 4:	Python code for 1EXP (-) Z+ Initial-Term Based Arithmetic Sequences formulation and weights coefficients generation function algorithm.....	158
Appendix 5:	Python code for the complete WVE brute exhaustive Optimization module	159
Appendix 6:	Python code for ROC Analysis.....	160
Appendix 7:	Python code for the WVE optimization data loading and Main Module.....	161

LIST OF ABBREVIATIONS AND SYMBOLS

$\operatorname{argmax} f(x)$	a value of x at which $f(x)$ takes its maximal value
\mathbb{Z}^+	Positive Integers
$(-)$	Negative
$1\text{Exp } (-) \mathbb{Z}^+$	1 Exponent Negative Positive Integers
$[\]$	Vector
$[\] [\]$	Matrix
2S-HHEC	2 – Stage Hybrid of Heterogeneous Ensemble Committee
AfSIS	African Soil Information services
AGDP	agricultural gross domestic product
AI	Artificial Intelligence
ALFbP	agricultural labor force by population
ANN	artificial neural networks
ANOVA	Analysis of Variance
ASMPSSA	Arithmetic sequences multi-precision search spaces
AUC	Area under the Curve
CIA	central intelligence agency
CNN	convolution neural networks
DBN	deep belief networks
DSR	design science research
DT	Decision Tree
E.I.A	extrapolated 1-acre harvest value.
EV	Evolutionary
EXP	Exponent
FAO	Food and Agriculture Organization
FN	False negative
FP	False positive
FPA	floating point arithmetic
FPN	floating point numbers
GA	Genetic Algorithm

GB	Gradient Boosting
GS	grid search
H/W	hardware
IEEE	electronics
I-T	Initial Term
K-Elbow	Knee Elbow
KNN	Knearest Neighbors
ML	Machine learning
NB	Naïve Bayes
RAM	random access memory
RF	Random Forest
ROC	Receiver Operating Characteristics
SFI	Soil Fertility Index
SIMD	single instruction multiple data
SISD	single instruction single data
SMAF	soil management assessment frameworks
SSA	Sub-Saharan Africa
SVM	support vector machine
TAMASA	Taking Maize Agronomy to Scale in Africa
TANSIS	Tanzania Soil Information services
TN	True negative
TP	True positives
WVE	Weighted Voting Ensemble

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

According to the United Nations Department of Economic and Social Affairs, Population Division (2022), as well as Food and Agriculture Organization (FAO, 2018), the world population is expected to increase from the current approximately 7.3 billion to an estimate of 9.8 approximately ten (10) billion people by 2050, as shown in Fig. 1. This entails the demand for increasing food productivity to ensure food security (FAO, 2018, 2017, 2016; Ishengoma & Athuman, 2018; Jayaraman *et al.*, 2016).

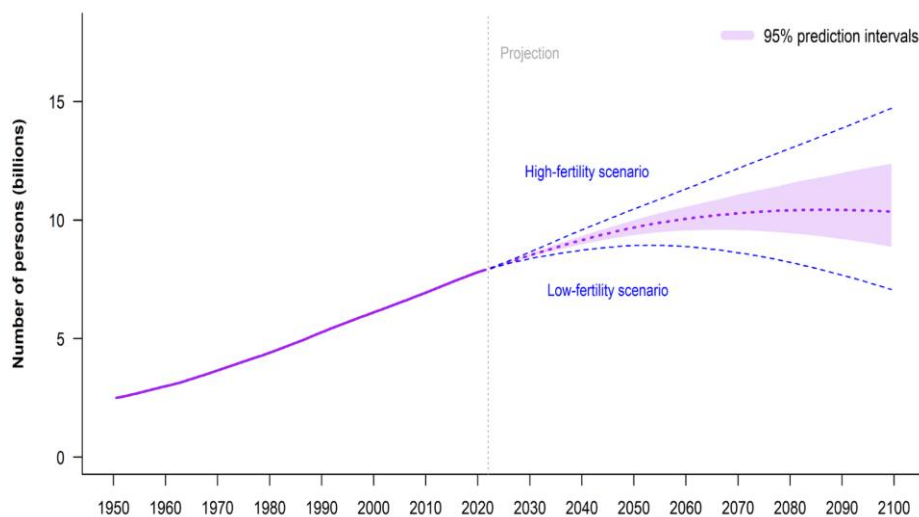


Figure 1: Global population size: estimates, 1950-2022

As such, efforts for improving crop productivity to facilitate sustainable agricultural intensification critically remain a key concern (FAO, 2017, 2018; Masri *et al.*, 2015), whereby fundamental key factors for crop production such as *soil* that harbors nutrients for crop consumption and growth to realize outstanding productivity (Kommineni *et al.*, 2018; Manjula & Djodiltachoumy, 2017; Rajeswari & Arunesh, 2016; Yusof *et al.*, 2016), is among the major focal point in that respect.

Essentially, that would require an understanding of the agricultural soil nutrients to analyze its fertility which can highly be validated by its resident chemical properties. Soil fertility exhibits variabilities in characteristics and has interlinked effects on crop productivity in terms of yield quantity (Hengl *et al.*, 2017; Ndakidemi & Semoka, 2006). It is important then to conduct soil

analysis through various approaches in the overall agricultural soil assessment frameworks in attempts to appropriately manage soil fertility for crop growth and improved yields. The assessment is advisably one of the critical best practices way before the 1970s, primarily with the involvement of the visual inspection method (Bünemann *et al.*, 2018). Figure 2 depicts a trend of Soil Assessment Methods through various methodological evolution.

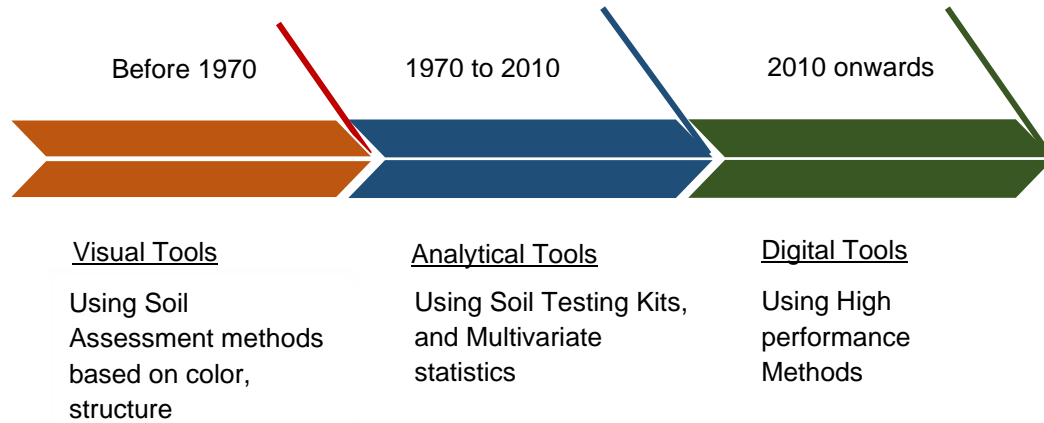


Figure 2: Soil Assessment Methods Evolution

While the visual soil fertility analysis was the earliest first approach before 1970, analytical methods that used multivariate statistical techniques came into play between 1970 to 2010 inclusive, and high-performance machine learning methods later proliferated starting 2010 to date (Bünemann *et al.*, 2018), studies were conducted from the 1980s involving machine learning applications whereby the vast amount of agricultural soil data can be harvested to derive valuable information in the form of trends and patterns. These can be harnessed to support optimal agricultural decision-making processes such as those geared towards improving crop productivity. Traditional human expert methods for gaining understanding from the collected data are limited, expensive, may overlook important details, and rarely may be subjective and biased (Gholap *et al.*, 2012a; Sirsat *et al.*, 2018).

Moreover, classical statistical analysis is often ineffective and inefficient with the increase in size and complexity of this data. In turn, contemporary machine learning (ML) algorithms, and data mining techniques for analysis through extraction and delivery of actionable information have been potential in solving various agricultural-related knowledge discovery problems ranging from *soil analysis and fertility status predictions* to compounds and other chemical properties (Jayalakshmi & Savitha, 2022), as well as crop diseases prediction, crop monitoring and predicting crop yields. ML modeling for soil fertility predictions is vital since the soil is

the key fundamental factor for crop growth and productivity, other than crop diseases, irrigation, weather, and climatic conditions management, as it harbors plantations and as a function of its fertility, it can supply plantations with the necessary nutrients as well as other chemical and biological possessions necessary for plant growth and increased crops yields, to eventually ensure food security (Manjula & Djodiltachoumy, 2017).

1.2 Statement of the Problem

Machine Learning (ML) based applications are one of the key solutions for sustainable agricultural intensification through the provisions of analytical information that is necessary to assist in effective decision-making processes such as for optimal soil fertility management to increase crop yields (Li, 2021; Menaga & Vasantha, 2022; Sharma *et al.*, 2020; Walter *et al.*, 2017). With that respect, strategies for improving decision-making in soil fertility management have been proposed in various studies (Gholap, 2012; Manjula & Djodiltachoumy, 2017; Massawe *et al.*, 2018; Sirsat *et al.*, 2018), whereby, ML techniques have been observed to widely be applied in agricultural soil modeling-related studies, with much exploration in their use to develop soil fertility status prediction classifier models by learning from the existent soil nutrients and their other key chemical properties.

However, the existing models for predicting soil fertility status have been observed to make use of a varying number of target classes in modeling soil fertility status classifiers and also demonstrated a varying range of predictive performances (Janvier *et al.*, 2021). While the use of varying target classes could be due to associated agricultural and modeling experts' subjectivity. The varying performances could stem from the involved ML implementation design and procedures, and soil qualities. Whereas, optimal target classes could be determined through the implementation of thorough ML approach designs. The theoretically and empirically widely appreciated weighted voting ensemble (WVE) scheme can be used to significantly improve individual soil fertility status prediction models' performance by ensembling their predictions into one predictive combination (Escorcia-Gutierrez *et al.*, 2022), due to their superior performances amongst other ensemble schemes.

Nevertheless, the WVE scheme is associated with the challenging task of searching and assigning the ensemble's base model optimal weights. This of which is another key significant factor for the WVE's ability to improve the performances through ensemble technique, other than heterogeneity or diversity in its constituting base model (Partalas *et al.*, 2008). While

currently these challenging tasks of WVE weights assignment can be achieved through advanced search procedures like the grid-based greedy search techniques such as greedy, and brute exhaustive search, among other variants. As well as, evolutionary optimization techniques such as genetic algorithms, differential evolution, evolutionary programming, and genetic programming. Contrary to the brute exhaustive search procedure (BESP), both the evolutionary and greedy-based search implementations are more efficient than BESP, but both may not guarantee to achieve of global optimum solution finding (Angulo *et al.*, 2021; Ast *et al.*, 2021; Bhspencer, 2015; Simon, 2015).

Therefore, the use of BESP which can produce optimal WVE models configuration sets with predictive performances similar to those created by evolutionary-based optimization techniques (Ariyanti *et al.*, 2019; Kurz *et al.*, 2020), and also can guarantee the finding of an optimal solution through a search across systematic search spaces, whereby the precision of the search spaces has been stated by Mouret and Clune (2015) to be the fundamental requirement for operationalization of search procedure towards finding the required optimal solution, may become imperative when the trade-off between the finding of an optimally accurate solution model to the time taken, that is efficiency, in finding that solution, becomes the main objective.

Therefore, this research aims to develop a high-performance soil fertility status prediction voting ensemble using brute exhaustive optimization in automated multi-precision of weights hybrid classifiers to theoretically guarantee the finding of an optimal weights solution, that is, those which provide for the most accurate WVE through a search across all possible candidate solutions combinations at a reasonable efficiency tradeoff.

The study first, identified agricultural soil properties to be attributed to soil analysis, and viable machine learning modeling algorithms for fertility status prediction. Then, an evaluation of the performances of heterogeneous ensemble models for predicting soil fertility statuses based on optimal soil fertility targets as indexed by crop yields was performed. Later on, the study developed and evaluated an optimal algorithm for explicitly generating systematic varying precisions weight coefficient matrices values as possible solutions search spaces as part of a brute exhaustive search procedure to optimize WVE for improving soil fertility status prediction performance, with accuracy maximization as the core objective function. Finally, an evaluation of the model's effectiveness in a maize field (experimental plantation) was done to evaluate the utility of the developed WVE model for predicting agricultural soil fertility status.

1.3 Rationale of the Study

On one hand, the use of varying fertility status target classes such as in Jayalakshmi and Savitha (2022) who used only 2 classes ‘Ideal’ and ‘Not Ideal’ as fertility classes to indicate the respective fertility statuses ‘low’ and ‘high’ may not capture the intermediary fertility status of medium soil fertility characteristics which may lead to the corresponding farm field site mistreatment with toxic over dosages if the fields location was considered low while it not, or may lower yield in case it was considered high while ws low or medium and left untreated appropriately. Rossel *et al.* (2010) used 3 classes ‘low’, ‘medium’, and ‘high’, while others used up to 5 fertility classes. On the other hand, the varying predictive performances amongst studies may have varying implications in the effective real-world applications of these models concerning the provision of reliable predictions. Thereby, this leaves room for innovation such as developing and implementing subtle design(s) for the determination of optimal fertility status target classes and approaches for model performances improvements.

On the other hand, most of the previous studies have not so far utilized solutions for ensuring the attainment of utmost optimal performances through the use of optimized WVE for predicting soil fertility statuses. Amongst other reasons, this is because while a greedy search is practically susceptible to the hill climbing problem which may lead to local extremums, and the evolutionary parent chromosome genes being the core of future fitter generations which may computationally at one time end up to un-fitter offsprings. Evolutionary procedures are inherent with a stochastic nature that may initialize WVE creature’s chromosomes genes that can at an instantiation probabilistically fail to can evolve fitter generations within the global context, hence theoretically they do not guarantee optimality (Team, 2021). Shown in Fig. 3 is the state of art search heuristics theoretical guaranteeing and non-guaranteeing of optimality tapping. As represented by the arrows, while, all three optimization techniques, that is, the brute exhaustive (which is indicated using the blue arrows), greedy, and evolutionary search techniques can be used to artificially optimize ML models, as depicted in Fig. 3, the brute exhaustive procedure can search across the entire set of possible combinations, whereby greedy search may not associate some of the combinations in the search due to the hill climbing problem which may intron return a suboptimal result, and evolutionary optimization implemented procedure’s chromosome initialization may provide for weak individuals that may not probabilistically not result into stronger selected current population parents whose offsprings may also not be fitter

parents to finally select the best solution from, resulting into suboptimal within the global evolutionary y search space context.

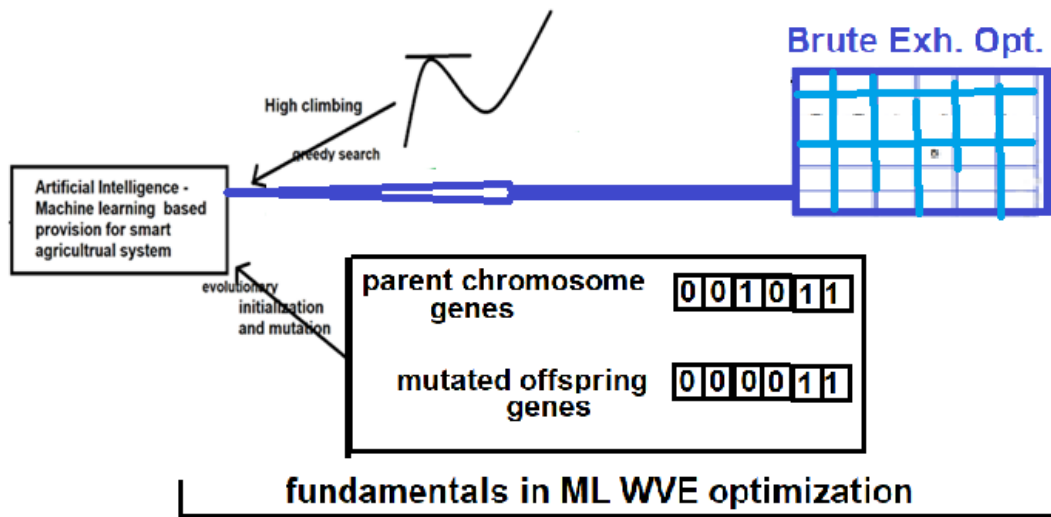


Figure 3: State of Art Search Heuristics Theoretical Guaranteeing and non-guaranteeing of optimality tapping

As such, as observed by Kurz *et al.* (2020), the surprising outstanding successes of the systematic brute force-based exhaustive search counterpart in producing optimal WVE models configuration sets with predictive performances similar to those created by evolutionary-based optimization procedures, in conjunction with its theoretical guaranteeing of finding the optimal solution through a search of all combinations or possible solution across a systematic search space can make it an imperative solution for attention in the tradeoff between high performances such as accuracy optimality tapping, and time of search across the spaces that have been clearly stated to be a key determinant factor for implementing a successful WVE optimal solution (Mouret & Clune, 2015).

Hence these facts bring the need for exploiting the search spaces to achieve the guarantee of optimality tapping. Whereas, in recent years, Hassanat *et al.* (2019) proposed an innovative deterministic approach to dynamically change crossover and mutation rates parameters for evolutionary-based GA solution space representations and parameter selection, the key cornerstone of their study being to coin the importance and use of solution search spaces formulation procedures of the GA's native operational mutation and crossover parameters interlinks and synthesis of their appropriate values ratios necessary for the implementation of effective evolutionary based generation population initializations for use in implementing an enhanced genetic based metaheuristics optimality search method. Nevertheless, there lacks an

exploitative exploration that emphasizes thoroughly on search spaces scrutiny as explicitly generated weights values as possible solutions for WVE optimization using computationally expensive brute exhaustive search heuristics, the method which has previously been described to have the ability to achieve optimization results similar to the evolutionary or greedy based search procedures, with the capitalization of guaranteeing optimality.

While this research aims to develop a high-performance soil fertility status prediction weighted voting heterogeneous hybrid classifiers ensemble which is optimized by using a novel mathematically represented multi-precision search spaces generation algorithm as part of a brute exhaustive search technique. The novel multi-precision search spaces generation algorithm as part of brute exhaustive searching will advance the body of artificial intelligence optimization or search algorithms knowledge. This of which prior lacked, the development of the novel varying precision and scales weights coefficients values matrix formulation algorithm as part of brute exhaustive search heuristic implementation procedure theoretically guarantees the finding of high-performance WVE models optimality through explorations that focus on search spaces inherent sizes or precisions and scales exploitation due to the significance thereof in the overall performance of a machine learning WVE model.

1.4 Research Objectives

1.4.1 General Objective

The general objective of this research is to develop a high-performance soil fertility status prediction voting ensemble using brute exhaustive optimization in automated multi-precision weights of hybrid classifiers.

1.4.2 Specific Objectives

The following are specific objectives:

- (i) To identify agricultural soil chemical properties for modeling soil fertility status prediction classifier models by using classical ML algorithms.
- (ii) To generate an effective automated multi-precision classifiers weights search spaces formulation algorithm for use in brute exhaustive optimization to guarantee optimal voting ensembles solution finding.

- (iii) To develop an optimal soil fertility status target classes prediction hybrid classifiers ensemble model through brute exhaustive optimization using the generated effective automated multi-precision classifiers weights search spaces.
- (iv) To evaluate the predictive performance of the resultant optimal WVE combination model against individual models, and competitive benchmark models.
- (v) To validate the WVE model predictions against soil laboratory base test results, and its utility in providing accurate soil fertility status predictions that can facilitate for application of appropriate remedies and soil fertility management practices, to improve maize crop yield as a case study.

1.5 Research Questions

The following were the research questions:

- (i) Which agricultural soil chemical properties can be used for modeling soil fertility status prediction classifier models by using classical ML algorithms?
- (ii) How can an effective automated multi-precision classifiers weights search spaces formulation algorithm to be used for guaranteed optimal voting ensembles solution using brute exhaustive optimization be generated?
- (iii) How can an optimal soil fertility status target classes prediction hybrid classifiers ensemble model through brute exhaustive optimization using the generated effective automated multi-precision classifiers weights search spaces be developed?
- (iv) How does the developed resultant optimal WVE combination model for predicting soil fertility status perform against individual models and competitive benchmark models?
- (v) How valid are the resultant WVE model predictions as compared to soil laboratory base test results, and does it provide accurate soil fertility status predictions that can facilitate for application of appropriate remedies and soil fertility management practices, to improve maize crop yield as a case study?

1.6 Significance of the Study

Analyzing soil information is one crucial requirement for the implementation of a successful smart soil fertility management system which is one of the key factors for sustainable agricultural intensification probably to meet the united nations Food and Agriculture Organization's demand for doubly increasing food productivity, in such to ensure the solution to food security by meeting the current and future projected population growth food demands. Whereas, soil data may provide generalized representations of patterns thereof to deliver useful information such as fertility levels, adequacy and accuracy thereof can be achieved through developing soil nutrients modeling and analysis using ML computational methods to provide a promising solution towards that endeavor.

The results of this research are crucial for the development of a smart soil fertility management system. These were obtained through the implementation of a high predictive performance WVE that can reliably predict agricultural fields' soils fertility statuses to provide optimal information about its resident fertility characteristics, thereby enriching the concerned practitioners with adequate soil and accurate information for future use, this being asserted by (Massawe *et al.*, 2018) as a key factor for improving soil fertility managerial decision making for improved food productivity.

Essential outcomes are described as follows:

- (i) To add to the body of existing knowledge an understanding of the state of affairs of machine learning techniques for modelling agricultural soils' key chemical properties predictive analytical applications.
- (ii) To advance or contribute to the existing ML WVE models optimization scientific knowledge body with the novel automatic weighting values generation algorithm function that particularly formulates multi-precision search spaces for a systematic brute exhaustive search heuristic implementation for optimizing weighted voting ensembles such that to guarantee optimality finding in with maximization of prediction accuracy performance as an objective function, in turn, improve the prediction performance of soil fertility status.
- (iii) Also, to contribute to the existing soil nutrients and other chemical properties hybrid modeling design for predicting soil fertility status at high performance using the optimal

number of fertility targets, in turn, provides for more site-specific fine-tuned predictive information.

- (iv) To provide an evaluation of ML model soil fertility status predictive performance, with a proposition of configuration set for the optimal ML WVE model.
- (v) To increase crop yields as a result of the use of the model-based predictions to gain a better understanding of the optimal soil fertility status and make better use of site-specific appropriate soil fertility management practices. This in turn may contribute to improved yield ratios, and bridge the agricultural digital divide which is amongst the key pillar of a sustainable smart global food supply system.

1.7 Delineation of the Study

Naturally, research studies are conducted under certain assumptions, and this study has no exception, some assumptions made in this research work are as follows:

- (i) This study is confined to machine learning-based classification of agricultural soils based on their features, the main reason for choosing soil over, irrigation, crop diseases, and other agricultural factors is the fact that agricultural soils are a key fundamental factor that needs addressing in a manner to understand of coexisting variabilities and manage them appropriately in order improve agricultural productivity and eventually ensure food security in general (Manjula & Djodiltachoumy, 2017).
- (ii) Whereas other soil chemical, biological, physical, and other sub-factors such as management practices or parameters that are significant to determine soil fertility such as climatic variabilities, are all important (Havlin *et al.*, 2016). In particular, this study is limited to soil chemical properties modeling with machine learning to develop an effective soil chemical fertility prediction classifier model for reliable predictions of future unseen soil samples, at high performance. Figure 4 depicts the study rationale with soil nutrients and chemical properties components, and provision of artificially intelligent programs for smart agricultural soil fertility management necessary for sustainable plant growth, using integration with ML application that is optimized for performance using brute exhaustive search. Whereby we are confided and particularly focusing on the soil's chemical properties due to other reasons. Firstly being identified as key fundamental parameters for the optimal determination and validation of soil

chemical fertility characteristic(s) concerning crop growth and eventual productivity, a fact that has been asserted also by Azhakarsamy and Sathiaselvan (2018). Secondly, machine learning-related research in the same objective has been studied in implementing models for similar objectives under a similar context herein in such to allow for performance comparisons with our improvement solution.

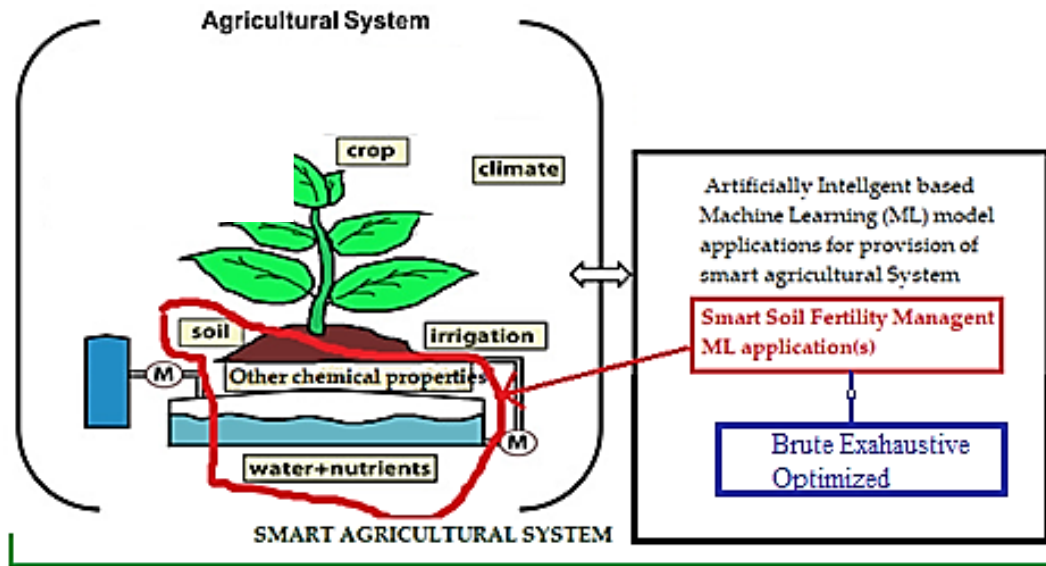


Figure 4: Depiction of the study rationale

This study, therefore focused on the identification and analysis of soil chemical fertility modeling features, targets, and corresponding ML algorithms, then optimal fertility class targets were engineered through modeling of the soil fertility status index as class targets, then an evaluation of performances of several classical ML algorithms was performed on the Tanzania nation soil chemical properties dataset. Then, the final high predictive performance model was obtained by combining the individual ML model results into one stronger judgment using an ensemble approach optimized using our novel proposed coefficients matrix algorithm was enhanced in performance. Finally, the best resultant model's utility was evaluated on a real field maize plantation experimentation to determine the model-based treatments' decision effects and end maize production results, with comparison against other existing non-model-based soil fertility treatment and management practice(s).

CHAPTER TWO

LITERATURE REVIEW

2.1 A Theoretical Literature Review

2.1.1 Agricultural Soils Properties for Fertility Status Analysis

Kommineni *et al.* (2018), Manjula and Djodiltachoumy (2017), Rajeswari and Arunesh (2016), and Yusof *et al.* (2016) asserted that agricultural *soils* are one of the key fundamental aspects of agricultural food productivity as they harbor nutrients for crop consumption and growth. Essentially, soil fertility is the ability of the soil to possess and supply necessary nutrients and other key soil properties for plant growth.

In agricultural soils in particular, there are key parameters that are functional to determine their fertility status and propose a remedy and management plan that are necessary to sustainably realize improved crop productivity (Bünemann *et al.*, 2018; Sirsat *et al.*, 2017). According to Bünemann *et al.* (2018), soil properties which include the soil physical, chemical, and biological properties should be selected based on some selection criteria that are bounded by conceptuality or operation ability, practicability or availability, sensitivity, and interpretability. And these properties are briefed as follows:

(i) Physical

Soil's physical properties are those which are associated with its natural morphology, amongst they include its structure: texture or type as clay, loamy, sandy, silty, peaty, chalky, or a mixture of these, which can now best be determined through the X-Ray capacity tomography 3-D images, water storage capacity, porosity, and infiltration (Emmet-Booth *et al.*, 2016).

(ii) Biological

These provide the key living component of the soil ecosystem, microorganisms that decompose organic matter into topsoil organic carbon responsible for the preservation of soil nutrients and formation of soil natural fertility; these can be measured in % to N, number of colonies (Ball *et al.*, 2017). Additionally, some other agronomic parameters are not part of the three soil properties categories but can be essential indexes for the estimation of soil fertility; they include crop type as seen in the work of Ball *et al.* (2017); and yield amounts (Sirsat *et al.*, 2017).

(iii) Chemical

Soil chemical properties are essentially those that are characterized by nutrients and other elements that are for direct consumption by plants. Often these are measured and used for the analysis of soil chemical fertility. These measurements of which can be determined through field trials for field samples by using several different soil testing methods to determine the availability and extent of different soil properties for different soil depths, mostly sub and upper (Ball *et al.*, 2017; Kavvadias *et al.*, 2018).

Most importantly, according to Bünemann *et al.* (2018) and Sirsat *et al.* (2017), soil fertility can be determined by chemical properties with which are mostly nutrients that can be determined by using either wet or dry chemistry through techniques such as Unmanned Aerial Vehicles (UAVs), near or mid infrared (NIR/MIR) spectroscopic method and calibrated using samples that underwent a wet chemistry method with reagents such as HClO_4 , HCl , HF and HNO_3 (Liu *et al.*, 2016), sodium hydrogen carbonate extraction (ISO 14263; ISO 1994), fulvic acids (FA), Humic acids (HA), BaCl_2 extraction (ISO 11260; ISO 1994), Dionex-100 Ionic Chromatography (DX 1-03, USA) (Emerson *et al.*, 1979; Kavvadias *et al.*, 2018), amongst others, as suggested by Bünemann *et al.* (2018), these soil chemical properties include total nitrogen content (N) in the form of nitrous oxide – N_2O , phosphorus (P) in the form of phosphorus pentoxide – P_2O_5 , potassium (K) in the form of potassium oxide – K_2O , sulphur (S) in sulphate – SO_4 form, iron (Fe), manganese (Mn), copper (Cu), zinc (Zn), boron (B), magnesium (Mg), calcium in calcium carbonate – CaCO_3 , cation exchange; organic carbon (OC) measured in percentage (%); electrical conductivity (EC) measured in milliSiemens per meter (mS/m) or deciSiemens per meter; potential hydrogen (pH) measured by a pH meter as acidic (0 to less than 5.5), neutral (5.5 to 7) or alkaline (7 to 14). Also other core soil chemical properties are the amount salt dissolved in water or saltiness, the salinity measured in ppm or %; and the amount of water contained in the soil, moisture, which is measured in percentage (%) of wet to dry soil (Bünemann *et al.*, 2018; Sirsat *et al.*, 2017).

2.1.2 Machine Learning Overview

Machine learning (ML) is a branch of artificial intelligence (AI) that has brought about many advancements in application areas such as robotics, natural language processing, and expert systems, and ML, is a subfield of computer science that focuses on the design and development of intelligent systems in the form of hardware, software, or both (Alpaydin, 2020; Baştanlar &

Özuysal, 2014). Whereas Machine engineering and learning is concerned with the design and development, and application of algorithms and techniques that learn by automatically organizing input data according to their common features to provide computer machine(s) with knowledge and experience in the form of mathematical models that infer the future with minimal human intervention thus with least errors and make decisions (Hurwitz & Kirsch, 2018; McQueen *et al.*, 1995; Mishra *et al.*, 2016; Mitchell, 1997), whereby ML does not involve consciousness as by humans, rather it statistical regularities or other data patterns, thus ML hardly resembles human approaches to learning (Ayodele, 2010).

Consequently, ML becomes one of the key aspects of an AI agent it provides for knowledge discovery of computational structures or models of an AI system, pattern recognition, and data mining by learning from data to discover computational structures of which can later on be used for predictive and descriptive work flows termed as improved performance analytics (Anifowose, 2020). Thus, the key takeaway of ML is when the performance of the machine improves with even a slight change in any aspect of an AI system then the machine is said to have learned.

Broadly speaking, a machine is said to learn when it changes its structure, program, or data, such that its performance in that carrying out artificial intelligence-related tasks that involve diagnosis, prediction, robotic control, planning, and recognition, improves with changes for enhancing its system or ab initio synthesis of new systems, as portrayed by Nilsson (1999), shown Fig. 5 is in the typical AI Agent System Architecture, with four (4) basic components of perception, model, action computation, and planning and reasoning.

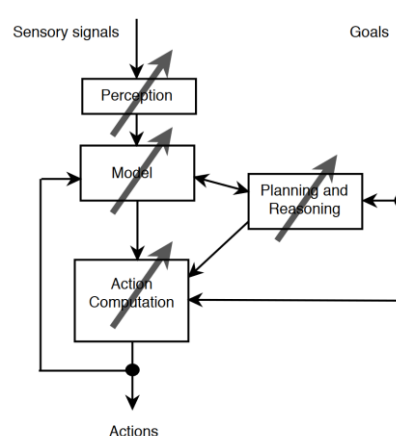


Figure 5: The AI Agent System Architecture

The agent perceives its environment, models it, and computes its action based on anticipated effects. Whereas, changes made to any of the components of the AI agent are considered learning, and the involved learning algorithm will depend on which component has been changed. While the key source of knowledge of an AI system is modeled data, the AI system would appropriately learn the required computational structures if the involved data or common sense is large. In addition, ML draws ideas from various including statistics, brain models, adaptive control systems, physiological models, artificial intelligence itself, and evolutionary models (Jh, 1975; Koza, 1994), also from the computational learning theory that provides for ML performance and computational analysis metrics, data mining, data analysis, data science, information systems, and computer science, among others (Han *et al.*, 2011; Tan, 2007; Witten *et al.*, 2016).

(i) Neuropsychological Learning Perspective

Based on the neuropsychological learning formulation called the Hebbian Learning theory, the prevalence of initial emergence in the field of machine learning was proposed in 1949 by Hebb (1949), the Hebbian Learning theory argument states that: *“Let us assume that the persistence or repetition of a reverberatory activity (or “trace”) tends to induce lasting cellular changes that add to its stability. When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that cell A's efficiency, as one of the cells firing cell B, is increased”*.

Machine learning for data science has been widely applied in various fields ranging from transportation, medical health, education, and agriculture, and has been reported to tremendously outperform humans as well as conventional computer programs performances by propelling very solid results in all tasks, for example, they could achieve approximately 99% accuracy, higher than humans at traffic signs (Golge, 2016). In medical health and education, machine learning has been relatively used for the detection of heart and breast cancer (Chaurasia & Pal, 2017); drug design (Burbidge *et al.*, 2001), and rational drug discovery (Zhang *et al.*, 2017); predicting student performance (Adhatrao *et al.*, 2013; Durairaj & Vijitha, 2014), and dropouts (Ameri *et al.*, 2016). Except for the four articles: a) “The Organization of Behavior New York” by Hebb (1949) which highlighted the initial emergence of core Hebbian learning theory, b) “Learning representations by back-propagating errors” by Rumelhart *et al.* (1988) which enlighten about the popular back propagation neural network (BPNN) machine learning method in existence long before the 1990's, c) “Induction of

decision trees” by Quinlan (1986) which proposed a remarkable discovery of the ID3 algorithm as a transparent and interpretable algorithms that can explain the underlying rules which are black-boxed in former algorithms and theories, and d) Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of development an expert system for soybean disease diagnosis by Michalski (1980) which highlighted on the earliest quotation in the application of machine learning in agriculture.

(ii) Machine Learning Algorithms

Machine learning has attained significant advances over the past three decades, starting early 90s with its algorithmic learning methods becoming the primary choice for practical predictive analytics software application development, among other they include automatic image classification or computer vision, natural language processing, sentiment analysis, spam detection, robot control, pattern detection, malware attack detection, diseases predictions, and automatic sequence processing for example in music or speech recognition, and other applications (Jordan & Mitchell, 2015).

The AI research has from its genesis been concerned with machine learning which can be achieved through the use of various algorithms or techniques that can make changes to the AI system model to learn through more experiences using these model learning algorithms that historically simply aim to learn some sort of computational structure, these include Functions, Logic programs and rule sets, Finite-state machines, Grammars, Problem solving systems, depending on the type of dozens of machine learning technique in existence and being used. For instance, learning input-output functions or simply learning functions computational structures were used as part of AI research, in 1959 by Samuel to develop a prominent early program that learned the parameters of a function for evaluating checkers game board positions (Samuel, 1959).

There are thousands of machine learning algorithms and hundreds more are being published each year (Brownlee, 2016; Castle, 2017; Domingos, 2012). In a broader view, these algorithms are categorized as shown in Fig. 6 of the non-exhaustive (Non-Exh.) taxonomy of ML algorithms, the learning computational structures or machine learning algorithms can be categorized into either supervised or unsupervised learning algorithm depending on the type of learning being conducted (Brownlee, 2016; Castle, 2017).

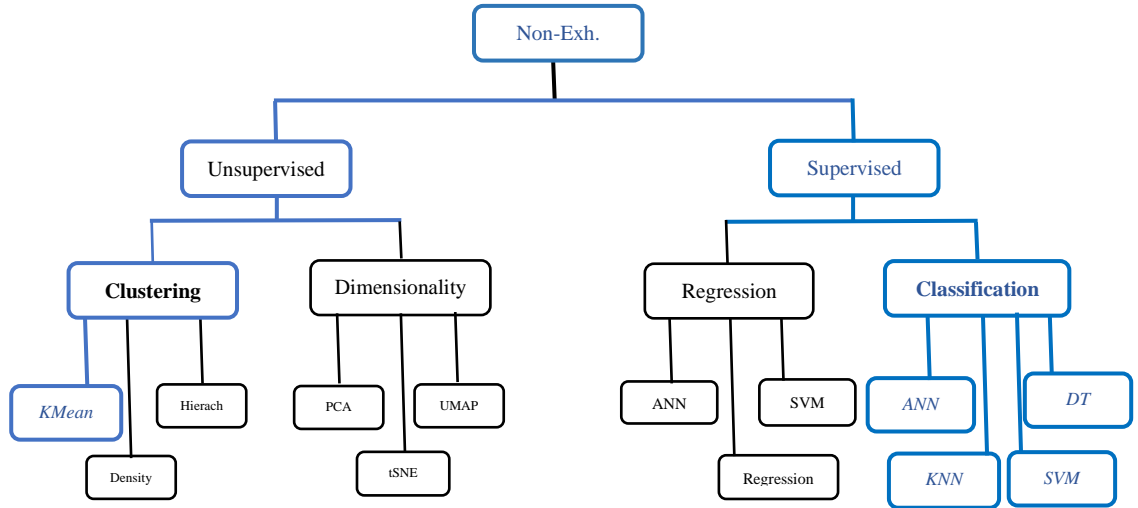


Figure 6: Non-Exhaustive Taxonomy of the different main ML algorithms

With focus to state of the art supervised learning algorithms, we base the description herein based on the profound Naïve Bayes (NB), Decision Trees (DTs), K-nearest Neighbors (KNN), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Bagging trees mainly Random Forests (RF), Boosting algorithms which we involve Adaptive Boosting (AdaBoost), and Gradient boosting (GB) (Ali *et al.*, 2015; Azhakarsamy & Sathiaselalan, 2018; Bhattacharya & Solomatine, 2006; Devi *et al.*, 2016; Kommineni *et al.*, 2011; Sirsat *et al.*, 2017). Most of the ML methods used today follow this workflow. Examples of such methods are artificial neural networks (ANN), decision trees, and support vector machines (SVM). Knowing what goes into the algorithms is essential to understanding how they work. The more we understand how they work, the more transparent they look, and the more we reduce the “black box” phenomenon that has been wrapped around them.

(iii) Supervised Machine Learning Algorithms

Supervised machine learning is characterized by a teacher or supervisor with the task to provide an agent, model, or function with a precise measure of its errors, whereby beliefs and common sense are presented in the form of a training data set made up of inputs and expected outputs or class labels are provided, and the function shall be used to infer for future unseen samples, thus the function will map a vector into a specific class from the several by looking at the functions input-output sets of examples (Ayodele, 2010; Osisanwo *et al.*, 2017). Generally, the traditional supervised learning algorithms use a training set D with variables constituting predictors X and target Y to train a model. The training process seeks to identify through an iterative procedure a set of model parameters that maximize the relationship between the

predictor (input features) and the target variables. The trained model receives new input data for the predictor variables and uses the recognized pattern to estimate the target variable. Figure 7 shows a simplified schematic of the traditional supervised ML workflow that has been in use for over 4 decades now.

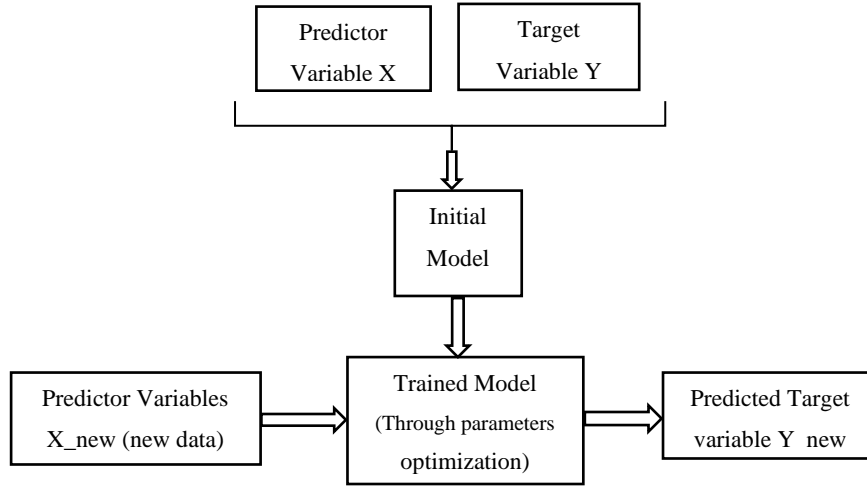


Figure 7: Traditional ML Workflow

For the case of a learning function computational structure, an ML algorithm will learn the input-output learning function f by using data D with n labeled examples having attributes $\{X_1, X_2, X_3, X_4, X_n\}$ that are to be used during training and testing on a set of unlabeled examples to determine the model's performance (Learned-Miller, 2014). Figure 8 illustrates the input-output learning function f , it is assumed that if a hypothesis h can be found, such that h closely agrees with f for the members of D , then this hypothesis will be a good guess for f especially if D is large like it was previously highlighted.

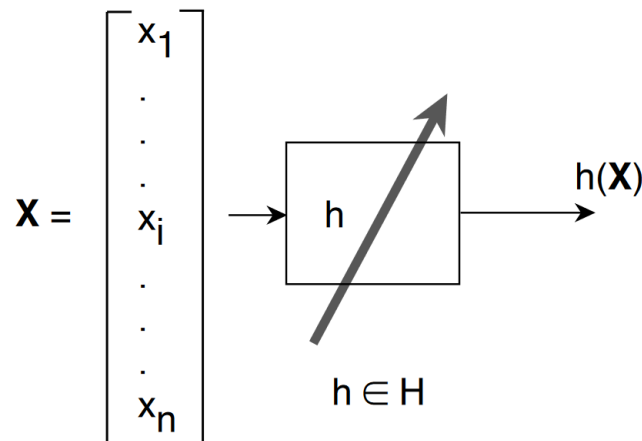


Figure 8: Input-output learning ML function

If the datasets used contains continuous values and created function produces continuous valued outcomes, the task is referred to as regression, otherwise, if they are based on a discrete number of possible outcomes, then it is a classification problem (Bonaccorso, 2017). Also, when the algorithm used to create the function is flexible enough and data is coherent, the overall accuracy increase, and the predicted to expected values difference close nearly to zero, the goal is to reduce the number of misclassifications and increase robustness to noise.

Through the application of various ML algorithms, supervised learning has been reported to be efficient in finding solutions to several linear and non-linear problems such as predictive analysis based on regression or categorical classification, robotics, sentiment analysis, automatic sequence processing such as speech or music, pattern detection, natural Language processing, plant control, spam detection (Sathya & Abraham, 2013), amongst others. Some of these profound ML-supervised learning algorithms are described as follows:

Naïve Bayes

A Naive Bayes classifier is one of the simple machine learning probabilistic classification techniques, which operates under the strong naive features independence assumptions, whereby it assumes inputs are independent of one another (Bhuyar, 2014). Figure 9 shows the Naive Bayes classifier represented as a directed non-acyclic graph network with each attribute being independent of each other (Nasteski, 2017).

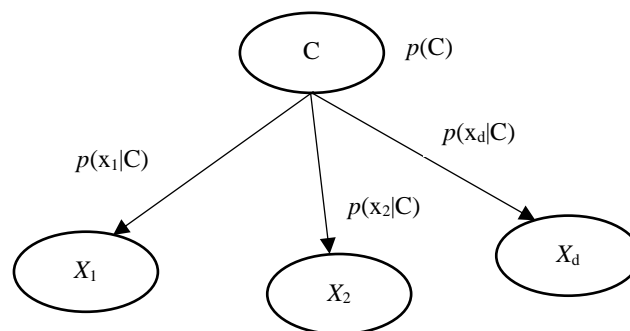


Figure 9: Graphical Network Graph Model Naive Bayes' classifier assuming independent input attributes

The Naive Bayes' graphical representation depicts how the possible de-naive Bayes' classifier dependencies namely, correlations, among the inputs are ignored to reduce or represent a multi-variate problem to a group of univariate problems. As a member of the Bayes family NB uses the theorem.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability
↓
Predictor Prior Probability
Posterior Probability

Whereby, for n predictor variable x_1 to x_n , it will be represented by the equation:

$$P(c|X) = \frac{p(x_1|c) * p(x_2|c) * \dots * p(x_n|c) * p(c)}{p(x_1|c) * p(x_2|c) * \dots * p(x_n|c)}$$

To learn the appropriate computational structure underlying the data. From the shown Bayes theorem, the posterior class probability $P(c|x)$ is calculated as the ratio between the product of classes' C prior probabilities $P(c)$ and likelihood of occurrences of n predictor variables $p(x_1)$ to $p(x_n)$ given the condition that the class is c , that is $[p(x_1|c) \text{ to } p(x_n|c)]$ multiply by $p(c)$, to the predictors' prior probability $p(x)$ alone. Whereby, with the Naïve Bayes algorithm only a small amount of training data will be required to estimate a classifier through probabilities of a given dataset instance, these of which are termed as class membership probabilities predictions, such as the probability that a certain tuple goes to a particular class (Jiang *et al.*, 2012; Kohavi, 1996; Kotsiantis *et al.*, 2007). Using each category prior probability given no information about an item, that is unseen data, naïve Bayes produces a posterior probability distribution over the possible categories described as an item.

K-Nearest Neighbors

Nearest-Neighbor (NN) is a memory-based ML method that can highly be related to statistical ones (Moore *et al.*, 1995). The algorithm works as follows: given a training set D with m labeled patterns, a nearest-neighbor procedure decides that some new pattern X which belongs to the same category as its closest neighbors in D .

More precisely, a k-nearest-neighbor method assigns a new pattern, X , to that category to which the plurality of its k closest neighbors belong. Using relatively large values of k decreases the chance that the decision will be unduly influenced by a noisy training pattern close to X . But large values of k also reduce the acuity of the method. The k-nearest-neighbor method can be thought of as estimating the values of the probabilities of the classes given X , and the denser the points around X , and the larger the value of k , the better the estimate. The distance metric

used in nearest-neighbor methods for numerical attributes can simply be the Euclidean distance described as the distance between two patterns $(x_{11}, x_{12}, \dots, x_{1n})$ and $(x_{21}, x_{22}, \dots, x_{2n})$, is $\sqrt{\sum_{j=1}^n (x_{1j} - x_{2j})^2}$, where n_j is the scale factor for dimension j (Friedman *et al.*, 1977). This distance measure can be modified by scaling the features to assume an almost similar spread of attribute values along each dimension. In addition, the distance calculations required to find nearest neighbors can often be efficiently computed by kd-tree methods (Friedman *et al.*, 1977).

An example of an 8 nearest-neighbors decision problem is shown in Fig. 10. In the figure the class of a training pattern is signposted by the number next to it, and a large number of training patterns must be stored to achieve its good generalization, the nearest-neighbor methods becomes highly memory intensive. Meanwhile, memory cost is now reasonably low, such that the method and its derivatives have seen several practical applications as seen in Moore (1991) and Moore *et al.* (1992).

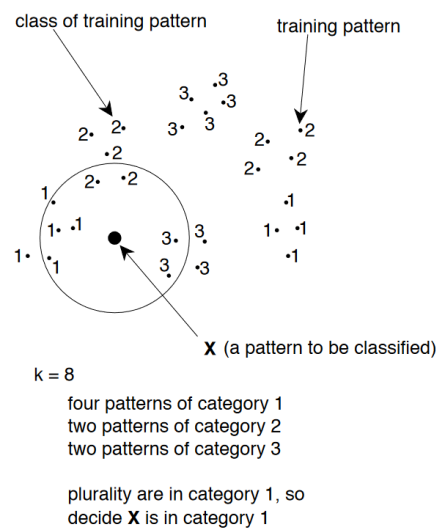


Figure 10: An 8-nearest neighbors decision problem

Decision Tree

Another remarkable discovery in machine learning was in 1986, when the ID3 DTs algorithm was proposed by Quinlan as an algorithm with the potential to provide transparent and interpretable explanations in the underlying rules, clearly stating reasoning behind reaching certain conclusions (Golge, 2016; Quinlan, 1986; Zhang *et al.*, 2017), who later on developed

further improvement of the ID3, the C4.5 and its Java version, J48 (Gholap, 2012; Melville & Mooney, 2003; Melville *et al.*, 2004).

Nilsson (1999) defined a decision tree as a tree whose internal nodes are tests on input patterns and whose leaf nodes are categories of the patterns. Shown in Fig. 11 is an example of a decision tree for assigning an input pattern to a class number or expected output by filtering the pattern down through the tests in the tree. Each test has mutually exclusive and exhaustive outcomes. For example, test T2 in Fig. 11 of the decision tree has three outcomes, whereby the left-most one assigns the input pattern to class 3, the middle one sends the input pattern down to test T4 for assigning to class 1 or 2, and the right-most one assigns the pattern to class 1. We follow the usual convention of depicting the leaf nodes by class number 1. Note that in discussing decision trees we are not limited to implementing Boolean functions they are generally useful for categorically valued functions.

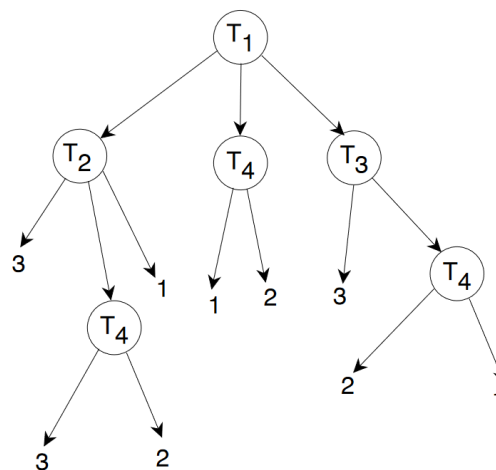


Figure 11: A decision tree

While as an advantage thereof, DTs are characterized by simplicity and comprehensibility in the determination and explanation of both small and large data structures, or attributes that provide the most information that can solve the classification problem and predict the required value (Ayodele, 2010; Jadhav & Channe, 2016; Mitchell, 1997). There are several dimensions along which decision trees might differ. One of the researchers who has done a lot of work on learning decision trees is Ross Quinlan. Quinlan distinguishes between classes and categories. He calls the subsets of patterns that filter down to each tip category and subsets of patterns having the same label classes. In Quinlan's terminology, our example tree has nine categories

and three classes. We will not make this distinction, however, but will use the words “category” and “class” interchangeably to refer to what Quinlan calls “class”.

Support Vector Machines

Subsequently, another great breakthrough in machine learning was the introduction of SVMs in 1995 (Cortes & Vapnik, 1995; Vapnik *et al.*, 1997; Zhang *et al.*, 2017), with its kernel version being released near 2000 making competition with the ANN community a bit more subtle, the SVMs could exploit knowledge of convex optimization, generalization margin theory and kernels against ANN, with stronger theoretical standings and empirical results (Cortes & Vapnik, 1995; Golge, 2016).

Depicted by Fig. 12 is an SVM discriminant function maximum margin which relies on maximizing the margin of error to select the best hyper-plane. The margin is determined by a set of hyper-planes parallel to the decision boundary on the positive and negative sides of the discriminant function each at the same distance to the boundary. When the margin is maximized, the training data points that are closest to the decision boundary are on the margin hyper-planes. These training data points are called the “support vectors.” Since the margins and the decision boundary are only determined by the support vectors, the SVM classification rule can be written as a function of these points. In practice, it is usually not possible to completely separate all training samples by a hyper-plane and some training samples can end up on the wrong side of the decision boundary or within the margin.

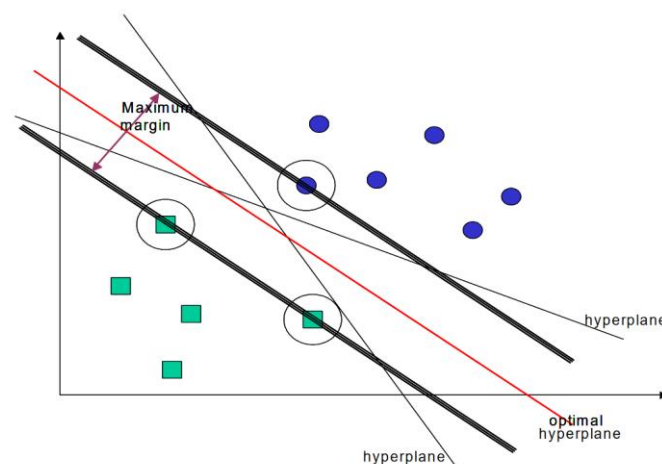


Figure 12: Support Vector Machine Maximum Margin

As a drawback of it, the learning algorithm of SVM often may select very few support vectors if the number of features encountered in the training data is usually small. In that case, SVMs are well suited to deal with learning tasks where the number of features is large concerning the number of training instances (Kotsiantis *et al.*, 2007). The SVM uses a complexity parameter denoted by C to control how much support points are penalized. A higher penalty means a more complex model with potentially more support vectors. The value of this parameter should be set using a validation dataset. SVM is capable of handling data sets having high dimensionality, *through* the mapping of points in space to create separate categories by maximizing the margin between different classes of points in linear problems (Poorinmohammad *et al.*, 2015; Zhang *et al.*, 2017), as it uses kernel mapping to transform nonlinear data sets into a high-dimension feature space that can be used in linear classification functions (Zhang *et al.*, 2017).

Artificial Neural Network

Long before the 90s, theories of conventional statistics existed along with core machine learning artificial neural networks (ANNs) and decision trees (DTs) basing techniques, and widely have been applied in different domains including agriculture and medicine as a means to supplement human expertise in the form of expert systems, as well as for educational purposes (Markoff, 1990; McQueen *et al.*, 1995; Zhang *et al.*, 2017).

Artificial Neural Networks (ANNs) which were largely created based on the neuropsychological learning formulation to mimic brain functioning an attractive and powerful model that was initially highly used in drug discovery research as of 1995; the ANNs topological structures could mainly be classified into four main approaches, namely, feed-forward neural networks (FFNNs), backward propagation neural networks (BPNNs), random neural networks (RNNs) and self-organizing neural networks (SONNs) (Zhang *et al.*, 2017), with the most popular BPNNs being suggested by Rumelhart *et al.* (1988), as a forward neural network with the multilayered perception that uses the gradient-descent method with the in the training set to minimizes the mean-square errors of the difference between the experimental data and the network outputs.

Generally, as shown in Fig. 13, an ANNs work by adjusting the weights of the network layer until the out of a function f that we try to find is the closest approximation of the networks output Y based on some input features $X = x_1, x_j, x_{k-1}, x_k$. Whereby the weights are incrementally increased slowly from the networks perceptron, neurons, or adilines of the TLU,

by using sigmoid units as initialization function to allow for continuous differentiability, whereby a value of 0 or 1 is to be assumed without the use of the regular TLUs which are linearly non-differentiable in such a function such as sigmoid was introduced (Anastassiou, 2022).

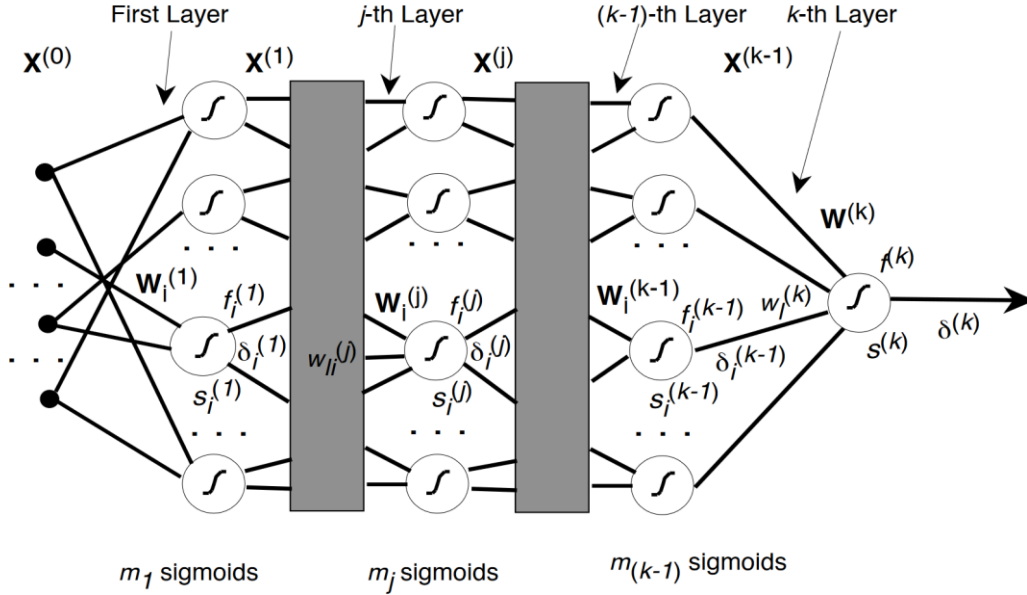


Figure 13: Artificial neural network layers with sigmoid function

The sigmoid function is superimposed in the traditional neuron networks threshold function to provide for differentiability and allow for carrying out of the network's equation partial derivative of f concerning its input so as convert the weights into a linearly differentiable function. The ANN portrayed in Fig. 13 above consists of one hidden layer, the more layers are added to that ANN, we would form a deep neural network (DNN) that will facilitate the purpose of deep learning through the data to fetch even more complex patterns otherwise not possible to obtain with unlike learning methods,

Deep Learning

Deep ML family of techniques has been reported to have the capability of achieving the highest performances than any other learning technique, including classical machine learning techniques and their variants (Kamilaris & Prenafeta-Boldú, 2018; LeCun *et al.*, 2015; Masri *et al.*, 2015). However, deep learning techniques are associated with drawbacks and limitations which are respectively high processing times and the need for very large image data (Masri *et al.*, 2015), while existing soil testing methods provide textual datasets contrary to images,

therefore deep learning cannot currently be applied for the analysis and prediction of soil nutrients and fertility status, otherwise we need to set up a project(s) for the collection of very large soil properties data in the form of images, with which this involves high cost and time also the tools are unavailable, except for the X-Ray capacity tomography which can only collect soil structural image.

In the context of agriculture on addressing soil nutrients analysis and fertility status prediction, based on the nature of agricultural soil data which is mostly textual, the techniques for learning can be supervised or unsupervised, with supervised being widely used for soil nutrients analysis and fertility prediction as observed from the literature. Whereas, ANNs determine and minimize errors through network adjustments (Ayodele, 2010; Pastur-Romay *et al.*, 2016; Siegelmann & Sontag, 1995; Yedjour & Benyettou, 2018), this gives ANNs an ability and edge advantages to detect all possible interactions between predictors variables without having doubts even in cases of the complex nonlinear relationship between independent and dependent variables.

In generally, achievements in discoveries of these new algorithms such as the support vector machines (SVMs) (Cortes & Vapnik, 1995; Marr, 2016); induction decision tree (ID3), a variant of native DTs (Golge, 2016); and random DTs forest which is commonly known as random forest (RF) algorithms (Breiman, 2001); is discussed in the way they could address the drawbacks observed in ANNs with forward and back propagations, mainly being due to the requirement of large amounts of computational times, especially if a lot of middle hidden layers are involved in the learning process the vanishing gradient otherwise termed as gradient loss or descent problem, that provides redundant learning hops after a certain period of time in such they are inclined to over-fitting in short number of hops (Hochreiter, 1991a); also due to ANNs limitations such as requirement of very large training dataset and black-box nature, that is, the inability to provide explanation of the underlying facts for reaching conclusions, sparked need for explorations by the machine learning research and development community (Siegelmann & Sontag, 1995), as a result these remarkable machine learning algorithms were discovered, including the said SVMs, ID3, and RF to encounter, among others, the mentioned problems of gradient loss, over-fitting, outlier susceptibility, black-box characteristics.

(iv) **Unsupervised learning**

Unsupervised learning algorithms are used to identify hidden patterns in unlabeled input data; they refer to provide the ability to learn and organize information without an error signal and be able to evaluate the potential solution, this type of learning simply models a set of inputs with no labeled examples (Ayodele, 2010; Bonaccorso, 2017). The lack of direction for the learning algorithm in unsupervised learning can sometimes be advantageous since it lets the algorithm look back for patterns that have not been previously considered (Sathya & Abraham, 2013). In unsupervised learning, training is conducted using dataset D without function and we aim to partition the training set D into subsets, D_1, \dots, D_R , in an appropriate manner. Whereby the value of the function is the name of the subset to which an input vector belongs.

In some cases when unsupervised results are to be used as inputs into a supervised process, problem domain expert(s) intervention becomes valuable for additional verification of the unsupervised learning intermediate results for enhanced performance and reliability, such as in the verification of different an unsupervised environment created clusters or otherwise termed as class labels to be used in supervised learning (Bhattacharya & Solomatine, 2006).

K-means

According to Taneja *et al.* (2012), k-means is one of the simplest unsupervised learning algorithms used to solve clustering problems by understanding and structuring data by grouping similar observations. However, the number of clusters must be specified in advance. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The k-Means algorithm is employed when labeled data is not available (Badillo *et al.*, 2020). The general method of converting rough rules of thumb into highly accurate prediction rules. Given a weak learning algorithm that can consistently find classifiers, as a rule of thumb, at least slightly better than random, with sufficient data. Figure 14 portrays the classical k-means clustering outcome with the value of $k = 2$. Every point is assigned to one cluster depending on the closest center point(s) marked by X.

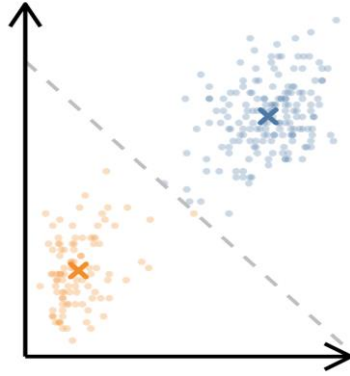


Figure 14: K-Mean clustering

No Free lunch theorem

Due to the no free lunch theorem and one does not fit all (Anifowose, 2020; Ho & Pepyne, 2002), these algorithms cannot apply and perform equally with similar performances in all problem domains areas, having different data with varying complexities, thereby they should be evaluated for performance and appropriate improvement methods including hybrid machine learning (HML), and ensembles, among others methods including improvement of the individual models' algorithm underlying computer structures and correspond mathematical theories and expression themselves, be developed to finally be able to create high-performance models. Table 1 presents a summary of ML algorithms' strengths and weaknesses.

Table 1: Summary of ML algorithms strengths and weaknesses

Algorithm	Concept behind	Strengths	Weaknesses
Naïve Bayes	Separate features based on probabilities	Simple and requires a small amount of training data	naive features independence assumptions
KNN	Distance (such as Euclidian) between the K closest neighbors	Simple structured	Memory intensive
SVM	Uses hyperplane and kernel trick on linear data	capable of handling data sets having high-dimensionality	suits learning tasks involving a large number of features concerning training instances
Decision Tree	Statistical	Simplicity and comprehensibility in the determination and explanation of both small and large data structures	Prone to over-fitting
ANN	Learns by adjusting network layers	detect all possible interactions between predictors variables Deals with non-linear data	inability to explain the underlying facts to reach conclusions requires of the very large training dataset inclined to over-fitting in a short number of hops
RF		robustness towards over-fitting and outliers large amounts of data No pruning, each tree is grown to the largest extent possible	

2.1.3 Hybrid Machine Learning

Hybrid ML (HML) is an advancement of the traditional ML workflow that seamlessly combines different algorithms, processes, or procedures from similar or different domains of knowledge or areas of application to complement each other. As no single cap fits all heads, no single ML method applies to all problems. Some methods are good at handling noisy data but may not be capable of handling high-dimensional input space. Some others may scale pretty well on high-dimensional input space but may not be capable of handling sparse data. These conditions are a good premise for applying HML to complement the candidate methods and use one to overcome the weakness or strengths of the others (Anifowose, 2020).

The HML algorithms are tailored towards the combination of two or more existing ML methods or combined methods from other fields such as the statistical domain, and HML methods have become common in recent applications. The HML algorithms are based on an ML architecture that is slightly different from the conventional workflow (Anifowose, 2020). We seem to have taken the ML algorithms for granted as we simply use them off the shelf, usually without considering the details of how things fit together. Whereas the possibilities for the hybridization of traditional ML methods are endless, new hybrid models can be built in different ways. Herein, we describe three basic types of HML namely architectural integration, model parameters optimization, and data manipulation which or along with any other relevant methods, can be used by ML fanatics to build hybrid models.

(i) The HML based on architectural integration

Architectural integration HML seamlessly wholly or partly combines the architecture of two or more traditional ML algorithms, in a complementary manner to evolve a more robust standalone algorithm. The most commonly used example is the adaptive neuro-fuzzy inference system (ANFIS) which is a combination of fuzzy logic and ANN principles (Anifowose *et al.*, 2013). Another example of an architectural integration HML method is the naïve Bayes tree which combines the architectures of naïve Bayes and decision tree algorithms. Whereby, the decision tree nodes would contain regular decision tree univariate splits, and the leaves contain naïve Bayes classification (Anifowose, 2020; Kohavi, 1996). Although decision trees can easily scale up to higher dimensional data, they are prone to overfitting. Whereas studies have shown the performance of the naïve Bayes algorithm to be excellent, naïve Bayes alone does

not nicely scale up under similar conditions. Therefore, a naïve Bayes decision tree hybrid method is necessary to leverage the complementary qualities of the two separate methods.

(ii) The HML based on model parameters optimization

Usually, to determine its optimal tuning parameters, a traditional ML method will use a search or optimization algorithm such as grid search or built-in gradient descent. Hybrid learning that is based on model parameters optimization seeks to complement or replace the built-in parameter optimization method by using advanced evolutionary algorithmic methods. For example, if the particle swarm optimization (PSO) algorithm is used to optimize the training parameters of an ANN model, then it can be referred to as a PSO-ANN hybrid method. In addition, a genetic algorithmic (GA) method that is used to optimize the ANFIS method training parameters would be termed a GANFIS hybrid model. The same goes with other evolutionary optimization algorithms such as Bee, Ant, Bat, and Fish Colony when in combination with traditional ML (TML) methods to form respective BeeTML, AntTML, BatTML, and FishColonyTML hybrid models.

(iii) The HML based on data manipulation

The HML which is based on data manipulation is probably the most implemented hybrid model(s). This type of hybrid learning seamlessly combines data manipulation processes or procedures with traditional ML methods to complement the latter with the output of the former. Whereby the simplest example is explained by the application of data transformation methods such as simple linear correlation analysis or principal component analysis (PCA) on our data before passing the data to the ML method, hence forming a hybrid computational structure.

Some practitioners use evolutionary algorithms to automate the optimization of the parameters of existing ML methods. The following examples are valid possibilities for this type of hybrid learning method, If a fuzzy ranking (FR) algorithm is used to rank and preselect optimal features before applying the support vector machine (SVM) algorithm on the data, this can be called an FR-SVM hybrid model. If a PCA module is used to extract a submatrix of data that is sufficient to explain the original data before applying a neural network to the data, we can call it a PCA-ANN hybrid model. If a singular value decomposition (SVD) algorithm is used to reduce the dimensionality of a data set before applying an extreme learning machine (ELM) model, then we can call it an SVD-ELM hybrid model.

The fuzzy logic method can be seen as a hybrid method if the fuzzification and defuzzification processes that come before and after the inference engine are respectively seen as kinds of preprocessing and post-processing tasks that are seamlessly integrated with the inference engine. Figure 15 shows a conceptual framework of the data manipulation HML workflow. These types of hybrid methods are often implemented in tasks that are based on feature selection, a type of data manipulation process that seeks to complement the built-in model selection process of traditional ML methods, which have become common. From studies by Anifowose *et al.* (2014), and Sasikala (2016) it was suggested the carrying out of this procedure using an external algorithm as a preprocessing step helps to complement the internal process by reducing the computational complexity, thereby increasing the accuracy of traditional ML algorithms.

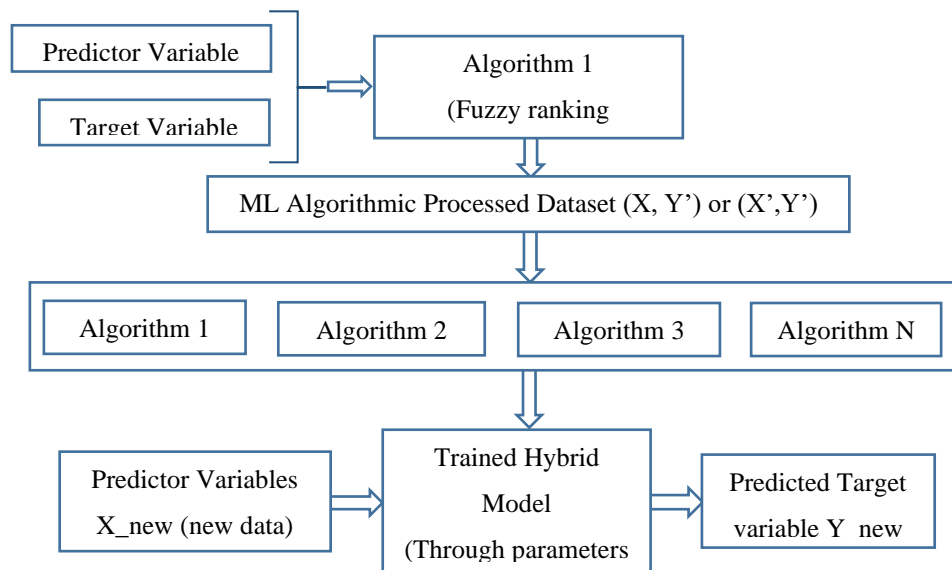


Figure 15: The Conceptual framework of a data manipulation HML workflow

2.1.4 Ensemble Learning

In ML, ensemble learning is a method used to combine results of various ML homogeneous or heterogeneous hypotheses or base experts' predictions that answer the same question to have more predictive accuracy (He *et al.*, 2017; Okey *et al.*, 2022; Zhou, 2009). Strategies for improving Machine learning (ML) model performance for optimal management decision-making have been proposed in various studies such as (Brownlee, 2016, 2018, 2020, 2021). Whereby, ensemble model construction has been implemented in several real-world applications due to their prospective superiority in performance as compared to single ML models, for that reason schemes thereof have widely been applied in several studies related to high-performance.

Machine Learning model implementations using ensemble learning strategies. One good example of ensemble learning superiority in performance over individual models was demonstrated by Dolzhikova *et al.* (2021), who implemented an ensemble model selection to integrate capabilities of CNN architectures and ensemble learning for decoding EEG signals collected in motor imagery experiments with achievement of over up to 7% as compared to individual models. For instance, in Ennoui *et al.* (2021) an ensemble learning scheme that used a weighted voting mechanism was implemented to combine VGG16, AlexNet, CNN, Inceptionv3, and mobileNet deep learning architectures for plant disease identification, whereby the weighting was performed using a genetic algorithm optimization based hybrid through G.A that along with initialization of weights, it may require extensive tuning as well to attain optimality. In machine learning, ensemble learning refers to the methods that use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the single learning algorithms alone (Deng *et al.*, 2021; Zhang & Ma, 2012).

2.1.5 Bagging

The basic principle is that a group of “weak learners” can come together to form a “strong learner”. Random Forests are a wonderful tool for making predictions considering they do not over-fit because of the law of large numbers. *Random forest (RF)* is a bagging ensemble learning algorithm, ML algorithmic discovery was initiated in 2001. The key characteristic of RF as an ensemble model is to combine individual decision tree models and provide for an improved model performance (Golge, 2016). As shown in Fig. 16 of the RF algorithm it uses random features selection and bagging concept to construct an ensemble of multiple DTs as base learners and train them using randomly sampled subsets of the original dataset, a consensus score is calculated as a weighted average or estimate of the individual DTs output to provide the final result” (Breiman, 2001; Zhang *et al.*, 2017).

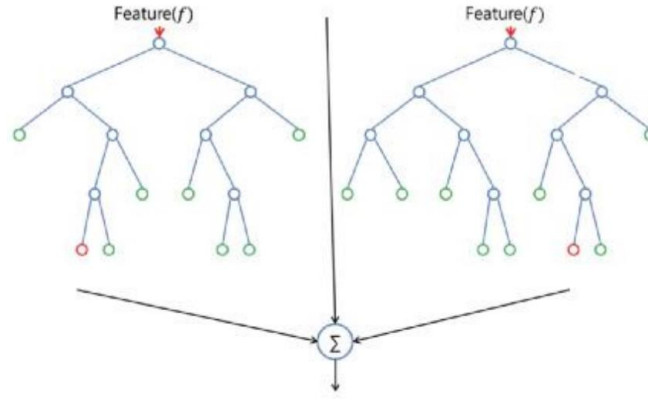


Figure 16: Random forest classifier

The key advantage of a random forest is its provision for more accurate and stable prediction through the construction and combination of several decision trees. Whereas it considers the most essential parameter while splitting nodes, by searching for the best features among random features subsets (Breiman, 2001; Kumar *et al.*, 2019). The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Assuming a training set with predictor variable X and targets Y where, $X = \{x_1, \dots, x_n\}$ with response classes $Y = \{y_1, \dots, y_n\}$. Then continuously for a definite number of times, bagging can be performed through selection from the training set, a random sample with replacement, and fitting a few hundred to several thousand trees depending on the size and nature of the training set of trees to the selected samples. Whereby, if the trees do not have any relation, the average of these trees is not so sensitive towards noise, unlike a single tree which is extremely subtle to noise inherent in the training set (Arooj *et al.*, 2018). The predictions uncertainty estimates can be made by evaluating class Y predictions standard deviations from samples X . whereas, strongly correlated trees can be created by training many trees on a single dataset. Bootstrap sampling is the use of different training sets for each of these correlated trees to de-correlate them. Random forest algorithm is capable of classifying large amounts of data with high accuracy without over-fitting, which makes it a wonderful tool for making predictions. Random forest trees are grown by random sampling with the replacement of N cases for a given training set. Whereby, the best split of the node is determined by the best split on m variables are selected at random out of the M which denotes the number of input variables. While the value of m is held constant during growing the forest, the tree grows to the largest extent possible without pruning (Breiman, 2001; Keerthan *et al.*, 2019).

2.1.6 Boosting

Boosting is generally a machine learning procedure to convert weak learners into a stronger model, two common types that exist among other includes adaptive boosting and gradient boosting. AdaBoost recognizes the weak learners' shortcomings through the use of weighted data points. The gradient boosting (GB) framework constructs additive regression models by sequentially fitting a weak classifier to current residuals (Friedman, 2001, 2002). As shown in Fig. 17 the architecture of gradient boosting previous weak classifiers' misjudgments are corrected to adaptively improve the overall prediction performance with high efficiency (Si *et al.*, 2017). The final model aggregates the results from all weak classifiers to achieve a "strong" classifier as an ensemble. In addition, Fig. 18 shows GB's gradient descent with a loss function to detect the residuals, such as mean squared error for regression or logarithmic loss for classification, eventually to improve its performance by minimizing the loss of weak learners to descend more towards a stronger ML model with minimal error as possible.

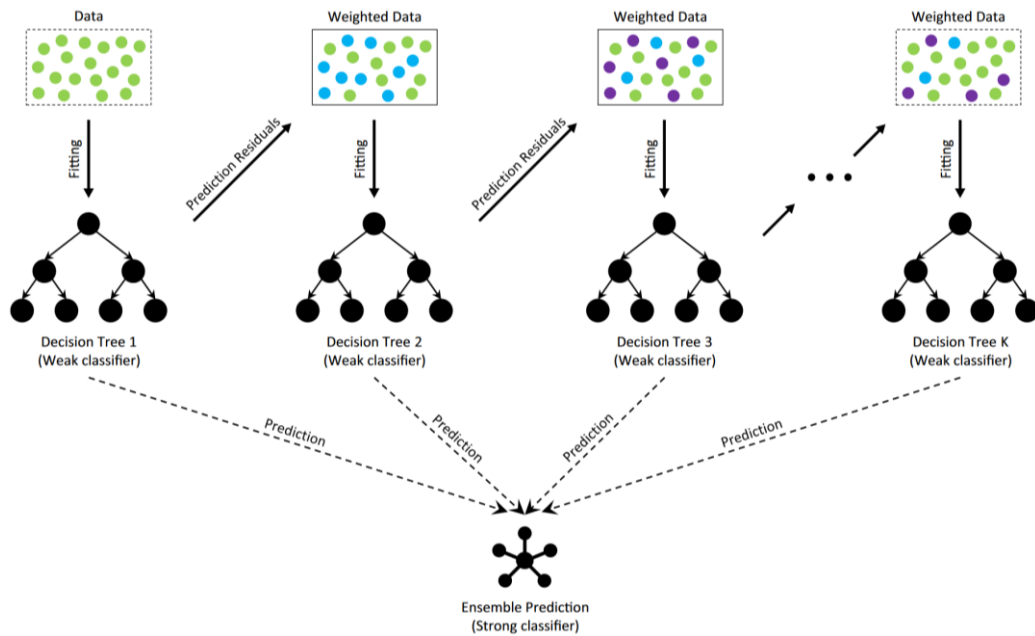


Figure 17: The Architecture of Gradient Boosting

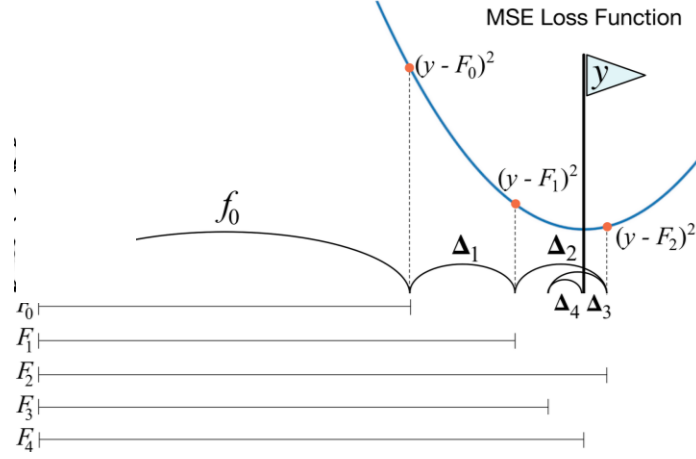


Figure 18: Gradient boosting's descent explanatory

2.1.7 Weighted Voting Ensemble Scheme for Model Performance Improvement

While several alternative implementations of ML ensembles could exist, model predictions or voting are the two common ways to combine single base model predictions to improve model performances, whereby variance and error reductions are capitalized (Pedregosa *et al.*, 2011). Voting selects the class that has mostly been predicted by individual models.

Generally, ML ensembles improve the predictive performances of individual learners by combining their predictions, through an ensemble function of all base members, and the ensemble error becomes a decomposition of average individual members' errors essentially to compensate for the lower average accuracy of individual members by the higher disagreement weight the ensemble as long as it is correct (Gomes *et al.*, 2017; Löfström, 2015; Pinellas & Livieris, 2020). *Weighted Voting Ensembles* (WVE) WVE models have extensively been appreciated due to their theoretical and empirical abilities to significantly improve individual learners' performance by treating each one of them as unequal and weighing them, contrary to its former variant, that is simple voting, which assumes all models to be equal (Dolzhikova *et al.*, 2021; Escorcia-Gutierrez *et al.*, 2022; Nuankaew *et al.*, 2022; Partalas *et al.*, 2008). Thus, as an improvement to simple voting, WVE whose final output is $y(x)$ can calculate as shown in equation (1)

$$y(x) = \operatorname{argmax} \sum_{i=1}^N w_i, jXA(Cj(x) = j) \quad (1)$$

“where y of all the unknown instances χ in the test sets are evaluated as the argmax function of the respective index with the largest value from array $A = \{1, 2, \dots, M\}$ denotes the set of exclusive class labels and χA indicates the characteristics function that considered the predictions $j \in A$ of a classifiers C_i on instances and create vectors where the j coordinates take

values of one and the remaining takes the value of zero (Dolzhikova *et al.*, 2021; Escorcia-Gutierrez *et al.*, 2022).”.

WVE was fundamentally introduced with this key understanding that different individual models to form an ensemble cannot in most practical cases have the same influence, thus treating them unequally and weighing their class probabilities prediction with unequal weights whereby the total sum of all models weights is equal to one (1) as represented in equation (2) (Brownlee, 2021; Dolzhikova *et al.*, 2021; Escorcia-Gutierrez *et al.*, 2022; Shahhosseini *et al.*, 2019; Zouggar & Adla, 2018).

$$\sum_{i=1}^k w_i = 1, \quad w_i > 0, \forall i = 1, \dots, k, \quad (2)$$

Where w_i is the weight of model C_k , whose total for all models 1 to k is equal to 1. Figure 19 depicts a WVE consisting of Logistic Regression, Random Forest, and Naive Bayes learners combined to achieve a more accurate ensemble learner for classification. From Fig. 19 of the schematic overview of the weighting algorithm for weighting algorithm, whereby the numbers are just pseudo, we see that a final prediction is obtained from combining Logistic Regression, Random Forest, and Naive Bayes learners to achieve a more accurate ensemble classification learner.

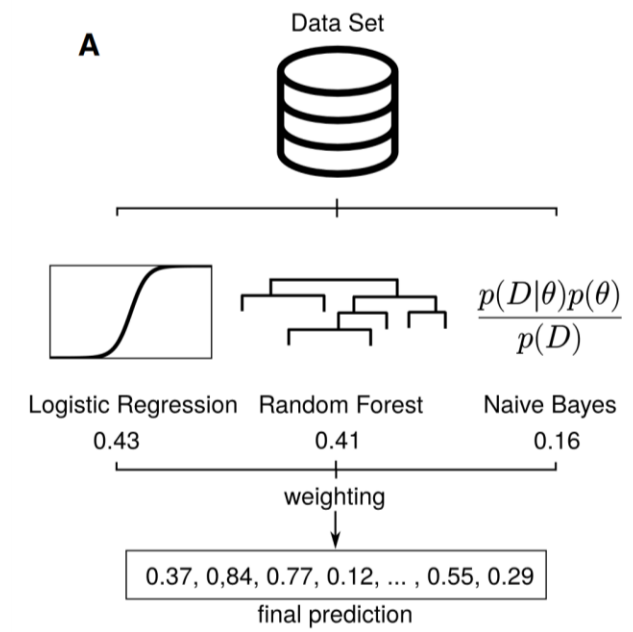


Figure 19: Schematic overview of the weighting algorithm

In particular, the weighted voting ensemble's output $y(x)$ can be expressed as in equation (1). Using a similar approach, much WVE development work has been done in numerous studies to combine ML models to improve their overall model predictive performance objective for a diverse of ML problem tasks. These include works by Wang *et al.* (2022) whereby a weighted voting ensemble method was used to obtain the improved output of prediction probability based on classifiers that predict the success rate of side-channel attacks according to the characteristics of side-channel analysis, whereby they applied the reciprocal of base classifiers success rate as a source for WVE assigned weights. In Escorcia-Gutierrez *et al.* (2022) an intelligent soil nutrient and pH levels classification and determination weighted voting ensemble deep learning system was proposed, whereby a wide range of simulations had to be carried out on a benchmark dataset to observe the performance of the WVE as it was being modeled.

In addition, an ensemble learning scheme that uses a weighted voting mechanism was implemented to combine VGG16, AlexNet, CNN, Inceptionv3, and mobileNet deep learning architectures for plant disease identification (Ennoui *et al.*, 2021), whereby the weighting was performed using a genetic algorithm optimization based hybrid through G.A that along with initialization of weights, it may require extensive tuning as well to attain optimality. Whereas Wu *et al.* (2021) and Ekbal and Saha (2011) used the genetic algorithm method to determine appropriate weights, the former developed a weight adaptation strategy to adjust base learners' weights based on their previous performances, and the later used GA chromosomes to encode weights of each of classifiers output classes. Ennoui *et al.* (2021). Also, Li *et al.* (2016) implemented a genetic algorithm-based search heuristic to find optimal weights of a WVE that was developed to effectively integrate twenty-five discriminative forecasted for piRNA prediction. Likewise, Zheng and Gu (2021) developed an ensemble model for classifying household solid waste via waste images based on CNNs architecture by using an unequal precision measurement weighting Strategy (UPMWS), that during model training, it capitalizes on the variations amongst the models' f1-score predictions performance to calculate the weights coefficients of their ensemble combination.

Last, but not least, Kurz *et al.* (2020) experimented with ensemble models for weighting scheme implementations based on the brute exhaustive, greedy, and genetic-based searching procedures, brute exhaustive was observed to produce an optimal model as effective as the counterparts at a high computational time of 23 hours.

2.1.8 Brute Exhaustive Search Algorithm

Nearly all science and engineering fields use search algorithms, which automatically explore a search space to find high-performing solutions (Mouret & Clune, 2015). The brute or exhaustive search algorithm is a set of instructions used to find the optimal solution by examining all possible solution combinations. This search process is not that new at all, it has been applied in several optimization problems including (Angulo *et al.*, 2021; Ast *et al.*, 2021; Dauzhenka *et al.*, 2018; Kaderzhanov *et al.*, 2021) to search for the most deemed optimal solution. Concerning WVE, the brute-force or exhaustive search algorithm has been used in various studies, like in Kurz *et al.* (2020).

Previous studies' results of brute exhaustive search depicting superiority in receiver operating characteristic (ROC) performance just like GA and Greedy Search, whereby as compared to linear model at significance with P less or equal to 5%, on a tested pima India diabetes correlated dataset. Figure 20 shows the performances of brute exhaustive search procedure in comparison to greedy search, and genetic linear search methods, to optimize individual learners' performances by weighting through them for the modeling of the Pima Indians Diabetes data set is another well-known data set that predicts diabetes mellitus in a high-risk population using diabetes dataset, where the brute force approach was slightly superior over the rest.

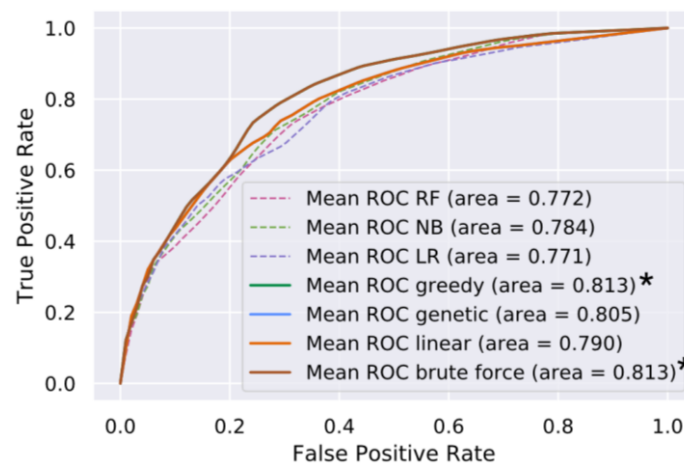


Figure 20: Previous studies' results of brute exhausted search exhibiting superiority just like genetics and greed

Table 2 indicates the computational times of the implementations, as reported by Kurz *et al.* (2020). It could be seen that the computational cost involved with a brute exhaustive search procedure is incomparably very high as compared to genetic and greedy search procedures.

Table 2: Genetic, greedy and brute experimental computational times

	SMR correlation	Computational time
Expert	0.578	NA
Brute Force	0.616	23 h ^a
Greedy	0.615	< 1s
Genetic	0.614	< 1s
QP	0.449	< 1s

In addition to that work, a brute exhaustive search was also implemented by Junk *et al.* (2015), and static and dynamic predictor weighting strategies were implemented and tested to improve the analog ensemble performance for wind power forecasting at on and offshore wind farms by using a brute force search procedure with error minimization over all possible predictor combinations. Furthermore, Abibullaev *et al.* (2020), did experiments on brute force based Electroencephalographic (EEG) signals as an architectural model for Brain-Computer Interface (BCI) research to enable individuals to interact with their environment by translating their mental imagery selection for convolution neural networks (CNN) to find for the parameters and provided results that are believed to suffice the verification of the efficacy for conducting a brute-force CNN model selection within a limited hyperparameter space, they examined whether a brute-force search with a limited space of hyper parameters for standard convolutional neural networks (CNNs) would possibly lead to a comparable classification accuracy as the state-of-the-art deep learning architectures for classification of motor imagery tasks we refer to any specific CNN that is constructed as part of the systematic model selection process as convolution network. Usually, the general basic algorithm that follows an exhaustive or brute force search requires two main stages: namely, Listing all the possible candidate solutions systematically, and checking for the optimal solution and reporting it (Angulo *et al.*, 2021).

While the main disadvantage of the brute exhaustive technique being its requirement for massive computational resources to find solutions in very large search spaces and which may sometimes make it slow and infeasible (Ariyanti *et al.*, 2019; Pedamkar, 2019), a drawback that can be addressed by using the search space reduction and algorithm parallelization strategies such as using parallel CPU–GPU computing structure, or computations and execution in a quantum environment. Its key advantage is the theoretical simplicity in implementation and ability to

always identify global optimal solution given computational resources are available (Okey *et al.*, 2022), which may make this algorithm be deemed a good choice especially when it will not require days, months, or years to locate the required solution in a real-life optimization problem.

Algorithm 1 next portrays a description of the basic exhaustive search steps as described in (Ariyanti *et al.*, 2019). Whereas, Algorithm 2 shows the brute force search procedure with strategies to speed up the algorithm evaluation as presented in Angulo *et al.*, (2021), where P is a valid problem's solution space, A is a null space, and c represents the candidate solution which is not null.

Algorithm1. Basic Exhaustive Search Steps

1. *Make a systematic enumeration of all possible solutions.*
 2. *Evaluate each possible solution one by one, and save the best solution time.*
 3. *Report the best solution.*
 4. *Taking care of large computational resource requirements.*
-

Algorithm 2. Exhaustive Search Algorithm with accelerator procedures.

1. *Search Space Reduction*
 2. *Perform Algorithm parallelization*
 3. *$c < \text{-first}(P)$*
 4. *while c is not equal to A do*
 5. *if $\text{valid}(P,c)$ then*
 6. *Output(P,c)*
 7. *$C < \text{-next}(P,c)$*
 8. *end while*
-

2.2 Empirical Literature Review

2.2.1 Application of Machine Learning Techniques in Modelling Agricultural Soil Nutrients and Other Chemical Properties for Fertility Status Prediction

Whereas, the earliest quoted example of the applications of machine learning in agriculture was in the use of similarity-based learning to identify rules for the diagnosis of soybean disease

(Michalski, 1980). Other studies of the machine learning field in agricultural problems include weather forecasting, yield prediction (Bagheri *et al.*, 2018; Devi *et al.*, 2016; Negied, 2014), fertilizers usage, fruit grading, plant diseases diagnosis and prediction (Michalski, 1980), pest management (Saini *et al.*, 2002), weed detection, soil nutrients analysis and fertility prediction for soil management and assessment (Azhakarsamy & Sathiaselvan, 2018; Kamilaris & Prenafeta-Boldú, 2018; Kommineni *et al.*, 2018), amongst others.

Studies related to soil analysis and fertility status prediction have widely reported that accurate soil fertility estimation and prediction models in agriculture can be achieved through the application of data-driven tools such as those using machine learning algorithms, as these can unlock such potentials. Sharma *et al.* (2015) asserted that the use of machine learning algorithms in this advent of large collected data for example can manipulate data and produce knowledge required for making better precision agriculture and support decision-making among farmers and other agricultural stakeholders, such as in soil assessment and management.

The genesis of ML methods in pedology traces back to the 1980s, when it was first applied in pedometrics whereby ML data-driven methods could be applied in the modeling and prediction of soil fertility. Presented here is a briefing on a few previous of these works from 2010 to date (2022). These works addressed a range of ML tasks from classifying soil properties of very low, low, moderate, high, and very high fertility status, to predicting unknown values. Whereas it is best practice to use as many possible algorithms, with all possible available principle parameters to perform an exhaustive evaluation to attain good analytical results and final model(s), Azhakarsamy and Sathiaselvan (2018) compared the performance of J48, KNN, JRip, NB, SVM, ANN classification algorithms by using PH, EC, N, P, K, OC, S, Fe, Mn, and Zn input variables of soil dataset to predict soil fertility as ‘fertile’ or ‘not fertile’, whereby JRIP scored maximum accuracy of 97%. In another study by Jayalakshmi R and Savitha Devi M (2022), data from Vellore soil testing laboratory with soil attributes PH, EC, Fe, Zn, Mn, Cu, OC, P, K, and fertility index (FI) as ‘ideal’ or ‘not ideal’ were utilized to perform experiments of training various bagging, boosting, and stacking ensemble classifiers, were they pre-processed the data, extracted relevant features as a means to achieve better performance, and attained an accuracy of 98.15% by boosting the decision tree like C5.0 algorithm.

A versatile method for rapid and accurate determination of soil fertility for sugarcane production was developed by Viscarra Rossel *et al.* (2010), whereby the soil fertility index was established and modeled independently using boosted decision trees with the use of soil

attributes PH, OM (OC), Ca, and Mg, Aluminium used in place of B due to their study finding a high correlation between the two, whereby they achieved AUC scores of 0.76, 0.67 and 0.65 for the respective fertility classes 'highly fertile', 'fertile', or 'least fertile' prediction. In another work, the Random Forest was used to develop a model that was used as part of the work to predict soil's OC, N, P, K, Ca, Mg, Na, Fe, Mn, Cu, Al nutrients fertilities and use the information to understand the edaphic drivers of soil constraints to very extreme high or near zero yields and heterogeneity across Africa, to guide in nutrients-specific interventions, they could find that soil factors could explain 72% of the variations in yields (Jin *et al.*, 2019).

Manjula and Djodiltachoumy (2017) developed a hybrid classification model by using a Decision Tree Classifier to isolate the soil's PH, EC, OC, N, P, K, S, Zn, Fe, Cu, Mn, and B dependent features and used Naïve Bayes classification on the independent features to predict the fertilities for the primary properties (PH, EC, OC, N) with individual naïve Bayes, and decision tree respective performances of 69.9%, 90.43%, and 99.93% for the DT-NB independent featured hybrid. While the macro P, K, S, and Zn, nutrients were respectively predicted at 38%, 88%, and 97% accuracies, the micro Fe, Cu, Mn, and B nutrients levels were predicted at 42%, 83%, 99.93% accuracies, respectively.

Kumar *et al.* (2019) examined soil micro and macro nutrients EC, K, pH, Mn, Zn, S, P, B, and OC using machine learning to grade soil nutrients, and they applied various classification algorithms and found that random forest had the highest accuracy score as compared to support vector machine and Gaussian naïve Bayes in predicting the soil classes for suitable crop plantation. Likely, Chaudhari *et al.* (2020) used PH, EC, OC, P, K, Fe, Zn, Mn, and Cu to implement machine learning models for predicting soil fertility as low, high, or medium using Support Vector Machine, nearest neighbor, Naïve Bayes, and Decision Tree that scored 60%. Also, Sirsat *et al.* (2018) implemented machine learning models for automatically predicting the Indian state of Maharashtra village-wise fertility indices of organic carbon (OC), phosphorus pentoxide (P₂O₅), iron (Fe), manganese (Mn), and zinc (Zn) by using 76 methods belonging to 20 families including neural networks, deep learning, support vector regression, random forests, partial least squares, bagging and boosting, quantile regression and generalized additive models, among many others.

Altogether, as per the Government of India's standard fertility levels, the prediction of nutrients fertility indices as low, medium, or high achieved the utmost best performance through the ensemble of extremely randomized trees (extraTrees), the results of which corresponded to

accuracy (Acc) and Cohen kappa values of (Acc= 86.45% Kappa= 69.60%), (Acc= 79.03% Kappa= 56.19%), (Acc= 79.46% Kappa= 52.51%), (Acc= 86.13% Kappa= 71.08%), (Acc= 97.63% Kappa= 81.03%) for OC, Fe, P₂O₅, Mn, and Zn, respectively, which is considerably fairly accurate. Other best-performing models were those generated through regularized random forests, random forests, and random forests with feature selection, last but not least good performances were obtained from gradient boosting of regression trees (bstTree) and generalized boosting regression (gbm); quantile random forest, M5 rule-based model with corrections based on nearest neighbors (cubist) and support vector regression (SVR). In another study, Escorcia-Gutierrez *et al.* (2022) designed an intelligent soil PH, OC, EC, P, K, B nutrient and pH classification using the weighted voting ensemble deep learning (ISNpHC-WVE) technique. Such classifications were employed in generating village-wise fertility indices analyses, and they are applied for making fertilizer recommendations using the decision support systems. In addition, three deep learning (DL) models namely gated recurrent unit (GRU), deep belief network (DBN), and bidirectional long short-term memory (BiLSTM) were used for the predictive analysis. Moreover, a weighted voting ensemble model was employed which allows a weight vector on every DL model of the ensemble depending upon the attained accuracy on every class.

Furthermore, Bhuyar (2014) used different classification algorithms to predict fertility rate based on soil's PH, EC, Fe, Cu, Zn, OC, P, K. Whereby, J48 classifier performed better in predicting fertility index for 6 classes very low, low, medium, medium-high, high, very high with 98.17% accuracy, while naïve bayes and random forest had respective performances of 77.18%, and 97.92%, their observation generally showed fertility rate for Aurangabad district to be medium.

In another study, Gholap *et al.* (2012) projected a comparative analysis of Naïve Bayes, JRip, and J48 ML algorithms by using soils data with attributes PH, EC, OC, P, K, Fe, Zn, Mn, Cu, it was observed that JRip classification algorithm gave better results compared to the other two algorithms, whereby it achieved an accuracy of 91.9% and therefore it was recommended to predict 6 soil classes very high, high, moderately high, moderate, low, and very low. Last but not least, a study by Massawe *et al.* (2018) was also useful in providing information on soil features, and algorithms of interest whereby PH, EC, N, OC, P, Ca, Mg, Na, K, Fe, Mn, Cu, and Zn could be observed key features these of which were modelled using naïve Bayes and

random forest trees as part of a task to numerically classify a portion of Kilombero Valley soil clusters in Tanzania.

2.2.2 Summary of the Empirical Review

Table 3 provides a summary of the reviewed studies related to the application of machine learning in soil chemical properties modeling.

Table 3: Summary of the State-of-the-art ML-based approaches and Soil chemical properties used in modeling nutrients and fertility status prediction

Author	Chemical Properties (features)	Dataset Size	Technique (ML algorithms)	Restraint(s)	
				Number of fertilities target classes	Max Accuracy/ROC Performance (%)
Azhakarsamy and Sathiaselvan (2018)	PH, EC, N, P, K, OC, S, Fe, Mn, Zn	127	J48, KNN, JRip, NB, SVM, ANN with 10FCV and % split	2 (fertile and not fertile)	97
Massawe <i>et al.</i> (2018)	PH, EC, N, OC, P, Ca, Mg, Na, K, Fe, Mn, Cu, Zn		NB and RF	Not applicable	Not applicable
Jayalakshmi and Savitha (2022)	PH, EC, Fe, Zn, Mn, Cu, OC, P, K	1430	TreeBag and RF ensemble bagging, C5.0 and Gbm boosting, KNN, CART, SVM, LR via GLM stacking ensemble	2 (ideal and not ideal)	98.15
Gholap <i>et al.</i> (2012)	PH, EC, OC, P, K, Fe, Zn, Mn, Cu	1988	NB, JRIP, J48	6 (Very High, High, Moderately High, Moderate, Low, and Very Low)	91.9

Jin <i>et al.</i> (2019)	OC, N, P, K, Ca, Mg, Na, Fe, Mn, Cu, Al		RF	Very extremely high or near zero yields	Not applicable
Author	Chemical Properties (features)	Dataset Size	Technique (ML algorithms)	Restraint(s)	
				Number of fertilities target classes	Max Accuracy/ROC Performance (%)
Rossel <i>et al.</i> (2010)	PH, OM (OC), Ca, and Mg, Aluminium were used in place of B due to their study finding a high correlation between the two	184	Boosted Decision trees	Class 1, the highly fertile soils; Class 2, the fertile soils; and Class 3, the least fertile soils	Class 1 in 75% of cases, Class 2 in 61%, and Class 3 in 65%
Chaudhari <i>et al.</i> (2020)	PH, EC, OC, P, K, Fe, Zn, Mn, Cu	Unidentified	SVM, KNN, Decision Tree, Naïve Bayes	3 (High, medium, low)	60%
Escorcia-Gutierrez <i>et al.</i> (2022)	PH, OC, EC, P, K, B	144	gated recurrent unit (GRU), deep belief network (DBN), and bidirectional long short-term memory (BiLSTM), and WVE	low, medium, and high. for each class, the pH level is divided into four classes strongly acidic, highly acidic, moderately acidic, and slightly acidic.	0.9281%, 0.9497% for PH
Manjula and Djodiltachoumy (2017)	PH, EC, OC, N, P, K, S, Zn, Fe, Cu, Mn, B	2948	Naïve Bayes, Decision Tree and Hybrid classification algorithm	5 (Very High, High, Medium, Low, and Very Low)	(PH, EC, OC, N) : 69.9%, 90.43%, and 99.93% (P, K, S, Zn) : 38%, 88%, 97% (Fe, Cu, Mn, B): 42%, 83%, 99.93%

Bhuyar (2014)	PH, EC, Fe, Cu, Zn, Mn, OC, P, K	1639	J48, Naïve Bayes, Random Forest	6 (Very low, Low, Medium, Medium high, high, very high)	98.17, 77.18, 97.92
Author	Chemical Properties (features)	Dataset Size	Technique (ML algorithms)	Restraint(s)	
				Number of fertilities target classes	Max Accuracy/ROC Performance (%)
Kumar <i>et al.</i> (2019)	EC, K, pH, Mn, Zn, S, P, B, OC	Unspecified	Random Forest Classifier, Support Vector Machine, and Gaussian NB	3 (Low, medium, and high)	72.74%, 63.33%, 50.78%
Sirsat <i>et al.</i> (2018)	EC, OC, N2 O, P2 O5, Fe, Mn, Zn, and B	930	NN, DL, SVR, RF, PLS, bagging and boosting, QR and extraTrees ensemble, Boruta, bstTree, and gbm; QRF, cubist, and svr.	3 (Low, medium, and high) per element or compounds	- OC (Acc= 86.45% Kappa= 69.60%) - Fe (Acc= 79.03% Kappa= 56.19%) - P2O5 (Acc= 79.46% Kappa= 52.51%) - Mn (Acc= 86.13% Kappa= 71.08%) - Zn (Acc= 97.63% Kappa= 81.03%)

From the empirical review based on previous ML-related soil nutrients modeling and fertility estimation research works. Some state-of-the-art ML Algorithms and Soil Properties could be identified. As shown in Fig. 21 of the Soil Parameters Use Frequency, it could be observed that the most used chemical properties include pH, as well as primary nutrients such as nitrogen, phosphorus, and potassium were mostly used. Also, electrical conductivity, organic carbon, and micronutrients such as iron, manganese, copper, zinc, and boron, were frequently used.

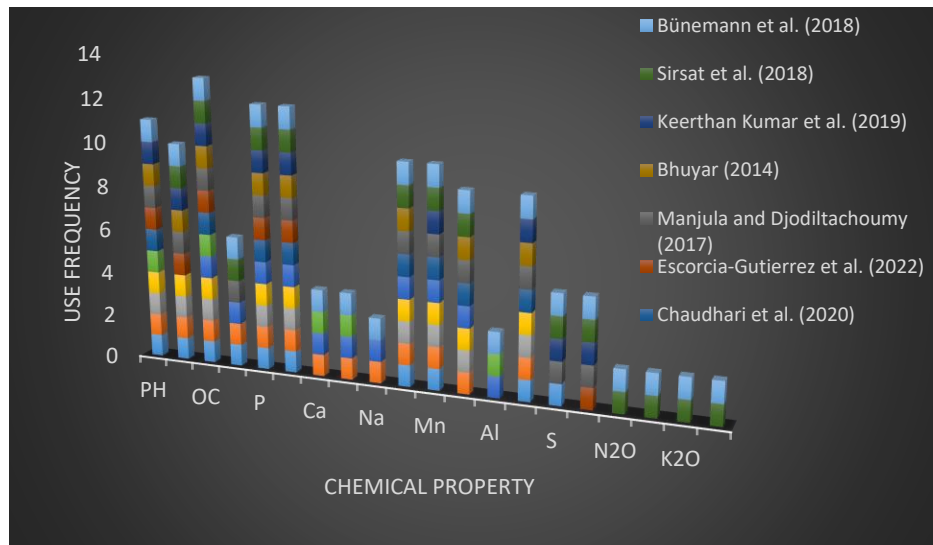


Figure 21: Soil parameters use frequency

As shown in Figure 22 the ML algorithms use frequency, and various algorithms used in previous studies could be identified. These are such as J48 (Java version of C4.5), Naïve Bayes (NB), JRIP, support vector machine SVM, artificial neural network (ANN), decision tree (DT), random forest (RF), and K-nearest neighbors (KNN). RF and NB are used most frequently followed by SVM, KNN, and J48.

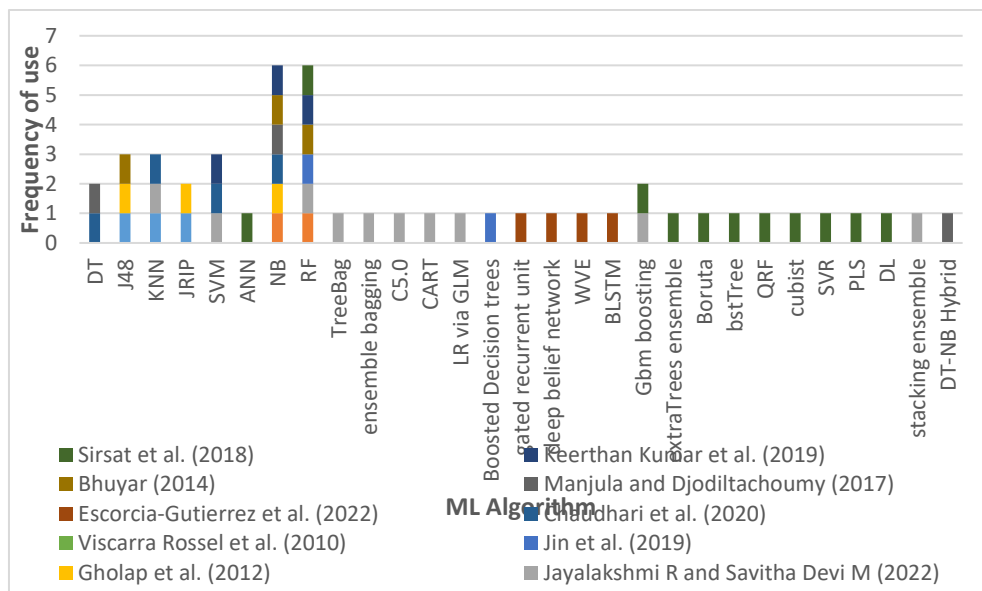


Figure 22: Algorithms use frequency

2.2.3 Research Gap

From the literature review, it could be observed that some of the studies used only two target classes low and high in classifying soil properties, and others used up to five target classes such

as very low, low, moderate, high, and very high fertility. Concerning ML algorithms employed in some studies, it could be seen that most used single ML algorithm model implementations. However, while the use of combination technique(s) such as the ensemble ML is one best method in improving ML predictive performance, there was a limited application of the technique in modeling soil fertility status high-performance prediction model(s).

On the other hand, given the challenges of implementing a deemed superior ensemble scheme, the WVE, the existent state-of-the-art procedures for searching optimal WVE solutions from the many existing in the solution search spaces are susceptible to various characteristics, with the greedy based being non-immune to the hill climbing problem, and evolutionary-based ones facing probabilities of non-optimality tapping due to the possibilities of not finding the fitter solution at the end of the search due to weak initialized parents that may never form the optimal solution during evolutions. Bearing in mind that search space sizes have been declared to be critical to the determination of significant search results and resultant high-performance WVE models, still, there is nowhere to be found a study that emphasizes exploitative scrutiny of search spaces on the resultant WVE model optimization by using the optimality guaranteeing brute exhaustive procedure. Even the existent limited ensemble technique application such as in the study by Jayalakshmi and Devi (2022) used homogenous ensemble committee members, let alone it did not apply the WVE scheme at all. As such all the predictive results from the models published in the literature generally portrayed varying predictive performances,

Therefore, there lack of a solution that incorporates heterogeneous prediction model considerations to reliably predict soil fertility status at high performance based on an optimal number of target classes, and. Eventually, this may respectively lead increase in incorrect predictions and imprecise application of fine-tuned fertilizer dosages according to predicted fertility status in the corresponding agricultural field's sites.

As such it may become imperative to develop reliable machine learning algorithms models to reliably predict soil fertility status at high performances. Mainly, this could be achieved by using the WVE method that is optimized by using a search procedure such as a brute exhaustive search technique that guarantees an optimal high-performance predictive model solution finding with emphasis on the search space exploitation to tap into the optimal weighting values sets for high accuracy resultant models realization.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

A research design is a key primary blueprint for conducting a scientific study in a tractable manner (Burns & Grove, 2010). In Design Science Research (DSR), as a meta-heuristic, the overall research approach or processes design is drawn by fitting together methods including strategies with data sources, case studies, data collection tools, and techniques, as well as an analysis to guide the researcher in conducting the study to derive answers from the research questions as explained by Suresh (2018).

Whereas, this study used a scientific approach, in both scientific and interpretivism research approaches, making an intervention that can affect organizational context (Giddens, 1986). The scientific approach assumes that phenomena can be observed objectively and rigorously well (Checkland, 1981), whereby originality and creative thoughts are required, and the research is sensitive to the scientist's psychological state (Wilson, 1990). Unlike Interpretive studies, where the understanding of phenomena is through meanings that are assigned by people (Orlikowski & Baroudi, 1991), with high involvement of subjectivity which is backed by qualitative arguments contrary to scientific bases of numerical exactness termed to statistical judgments as highlighted by Garcia and Quek (1997). This research used positivism research philosophy which involves objectivity and is backed by quantitative arguments.

3.2 Design Science Research

The DSR is a meta-heuristic method well suited for information system-related research studies (Hevner & Chatterjee, 2010a; Venable, 2006), as it provides a blueprint of a flexible way to scrutinize an organizational situational analysis for its political, economic, social, technical, legal, environmental (P.E.S.T.L.E) strengths, weaknesses, opportunities, and threats situational analysis. The DSR methodology constitutes three (3) iterative cycles or phases that form the heart of this provides Meta heuristic method. As shown in Fig. 23, these involve the relevance, design, and rigor cycles whereby the organizational application domain requirements concerning any one or a combination of the P.E.S.T.L.E perspective can raise a call for a DSR artifact development.

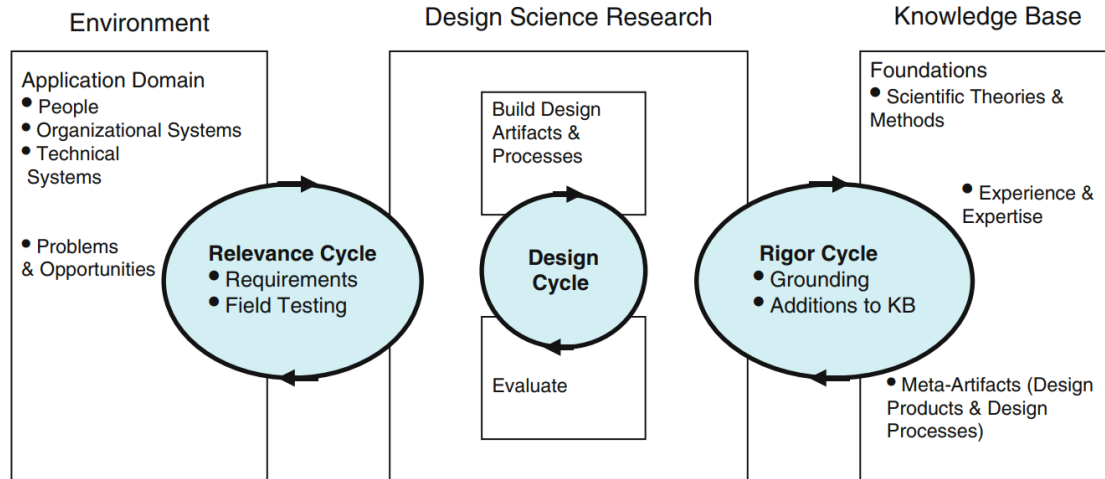


Figure 23: The DSR methodology cycles

This methodology is recently been highly appreciated as a blueprint for information technology and communication-related research design and development. Among the major potential of its use is its appropriateness in solving organizational research activities with explicit emphasis to base on rigorous grounding scientific theories and methods, experience and expertise, design products, and design processes. And these are highly expected to coexist in the existing DSR knowledge base, otherwise, new knowledge would require to be developed.

The DSR mainly is geared towards finding suggestions for the identified organizational problem(s), to developing the relevant solution as artifacts that can be benefited from their utility as part of a DSR solution that should finally be communicated back to the scientific body of knowledge (Baskerville *et al.*, 2009, 2018; Gregor & Hevner, 2013; Järvinen, 2007; Vaishnavi & Kuechler, 2015; Vaishnavi & Kuechler, 2004, 2016, 2017). These artifacts can be such as constructed symbols and vocabularies, model representations and abstractions, algorithmic methods, implementations, or system prototypes, as well as theories (Hevner & Chatterjee, 2010b; March & Smith, 1995; Venable, 2006).

Hevner *et al.* (2004) placed forward two fundamental questions of DSR: (a) "What utility does the new artifact provide?" and (b) "What demonstrates that utility?". Hevner *et al.* (2004) stressed that evidence must be presented to address the two questions and that contribution arises from utility as a result of the inventiveness in discovering the solution to the recognized problem. Peffers *et al.* (2007) recommended that the development of the artifact should be a search process that draws from existing theories and knowledge to create a solution for a

defined problem. The solution of which should be evaluated and validated as an artifact by using both existing and new theories.

In this research, the general methodology for design science research was adopted to develop a novel machine learning modeling design for implementing a reliable soil fertility status prediction performance improved model, as a contribution artifact, to the body of brute exhaustive search and machine learning WVE optimization knowledge, new 1EXP (-) Z^+ based brute exhaustive search algorithm for improving performance of base models through an optimal WVE combination, was developed by drawing from the DSR knowledge based on existing knowledge/theories. The DSR deemed fit to set the research design which can provide developers an implementation roadmap for building the utmost comprehensive computational or rather ICT-related solutions in general. Explained by the DSR's 6 activities that were studied and synthesized by Peffers *et al.* (2006), namely: (a) problem identification and motivation, (b) objectives of a solution, (c) design and development, (d) demonstration, (e) evaluation, and (f) communication, these of which were aligned with the specific research objectives and corresponding implementation methods as follows:

3.2.1 Problem Identification and Motivation

In this step, the study focused on reviewing literature from various existing research work. The purpose of reviewing the literature was to create a clear understanding of the research problem, build a theoretical knowledge base related to specific research questions to be answered by this study, and determine the scope of the study.

3.2.2 Define the Objectives of a Solution

To define the objective of the problem solution that was generally identified in the first activity, a more focused literature review was conducted to set the research objectives, and the direction forward. As shown in Fig. 24, research works related to ML application in soil fertility, WVE optimization, GA, and greedy-based brute exhaustive search were reviewed.

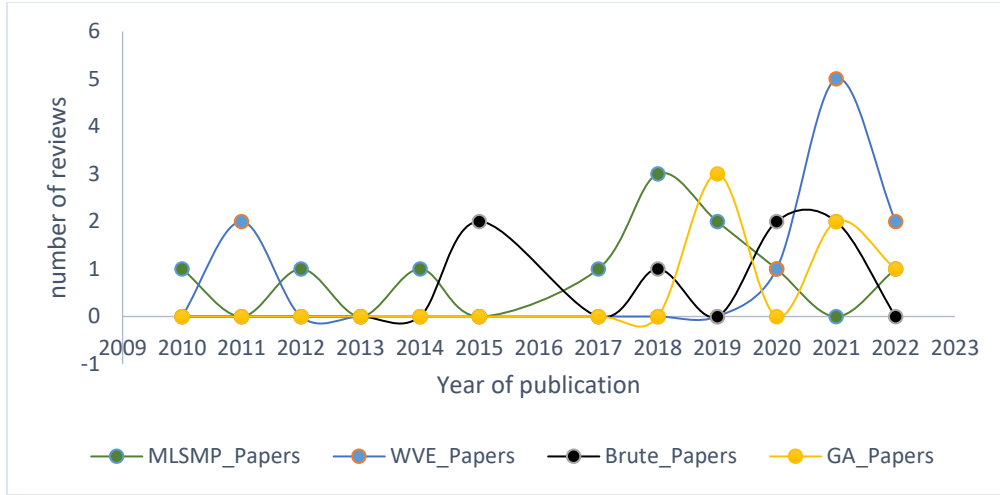


Figure 24: The ML and optimization-related reviewed papers publications

3.2.3 Design and Development

During this step, the focus was on the implementation of the hybrid classifiers, as well as designing and developing the brute-exhaustive search procedure for optimizing the WVEs of the hybrid classifiers. Machine Learning Modelling Algorithms were rigorously applied as they are key methodological implementations for the problem stated herein.

The study used brute exhaustive optimization to develop the search procedure for finding the proposed optimal ML WVE technique-based model. The search procedure was mathematically implemented by using linear algebra vectors and matrices of probability predictions, vs multi-precision weights values which were automatically generated by a novel arithmetic sequences-based search space generation exponential function.

3.2.4 Demonstration

Demonstration of the performance of the developed solutions is a key ingredient of DSR, whereby the DSR artifacts are tested for performance. This study performs laboratory simulations of the model to gain an understanding of the developed model's artifacts' performance using relevant evaluation ML model metrics. In addition, the evaluation of the utility of the developed artifact was conducted through real-world model-based recommendations for decision-making on any required soil fertility deficiencies before plantation, in a maize field's plantation experimentation. While the effectiveness of the algorithm in the significance of search spaces in optimization was further demonstrated by using differences in represented model accuracies.

3.2.5 Evaluation

To evaluate our proposed artifacts, the rigor ML models evaluation metrics derived and present in the literature were used, these of which were obtained from the DSR ML models performances evaluation metrics knowledge base, while the effectiveness of the algorithm in the significance of search spaces in optimization was further evaluated by using the accuracies plotting curves differences profiles. Finally, maize yields per acre were used to measure the effect of model-based soil fertility prediction recommendations to guide the decision-making of where and how much to treat the soil.

3.2.6 Communication

The results and findings of this study were effectively communicated to technical and managerial audiences through journal publications, conferences, workshops, seminars, and poster presentations.

3.3 Materials and Methods

3.3.1 Study Area

Tanzania's southern highlands, Uyole in the Mbeya region was used as an experimentation site, as Mbeya is one of the major staple crops production regions in Tanzania, similar reasons apply to the choice of maize crop for experimentation (Rurinda *et al.*, 2020). Figure 25 shows the Mbeya study site in Tanzania where the studies developed model was validated for utility in Tanzania.



Figure 25: Study experimentation location

3.3.2 Modeling Dataset

The dataset used to develop the models and associated analysis was secondary data which were obtained from the Tanzania Agriculture Research Institute (TARI), under the African Soil Information Services (AFSIS). Maize yield data was gathered from and Tanzania Ministry of Agriculture Statistical Division. Table 4 shows a description of the agricultural soil properties and maize yield dataset features.

Table 4: Description of the agricultural soil properties and maize yields dataset features

Attribute	Description	SI Unit
OC	Organic Carbon	Percentage
pH	Potential Hydrogen	Neutral, Acidity/Basicity degree
EC	Electrical Conductivity	deciSiemens per meter
TN	Total Nitrogen	Percentage)
P	Phosphorus	Milligrams per kilogram
Ca	Calcium	Centimoles per kilogram
K	Potassium	Centimoles per kilogram
Mg	Magnesium	Centimoles per kilogram
Na	Sodium	Centimoles per kilogram
S	Sulphur	Milligrams per kilogram
Mn	Manganese	Milligrams per kilogram
Al	Aluminium	Milligrams per kilogram
Zn	Zinc	Milligrams per kilogram
Fe	Iron	Milligrams per kilogram
B	Boron	Milligrams per kilogram
M_Yld	Maize yields	Tons per Hectare numeric

As can be seen, the dataset contained 16 features, 15 of which are the key soil chemical properties necessary for the determination of fertility level as defined in Bünemann *et al.* (2018), and that has been utilized by studies mentioned in Section 2, the dataset as well contained the corresponding maize yields in harvested tons estimates mapping as index to soil

fertility status. Acquisition of data required for ML modeling is a key step towards the overall development procedure. Therefore, two (2) different sets of agricultural soils dataset match the previously described modeling data. The first was 6260 instances of Tanzania soil data which was used for training and testing the model, and this came from TARI.

Table 5 presents a portion of the TARI and Tanzania Ministry of Agriculture's respective Agricultural Soils Raw Data with corresponding maize grain yields.

Table 5: Tanzania Agricultural Research Institute and Tanzania Ministry of Agriculture's respective Agricultural Soils Raw Data with corresponding maize grain yields

SSN	OC	pH	EC	TN	P	Ca	K	Mg	Na	S	Mn	Al	Zn	Fe	B	Yields
2015 TanSIS_TOP-0a8Q4cYG	0.68	5.87	0.001	0.02	14.58	0.054	0.016	0.034	0.788	28.102	6.04	48.06	0.828	74.55	0.010	1.21
2015 TanSIS_TOP-0BSeWh1w	1.28	6.09	0.001	0.04	3.600	0.058	0.018	0.006	0.054	28.808	5.49	55.63	0.437	18.88	0.020	1.11
2015 TanSIS_TOP-1Q94pxhk	1.31	5.66	0.001	0.04	1.270	0.070	0.034	0.009	0.008	46.138	4.36	48.76	0.371	31.77	0.010	1.32
2015 TanSIS_TOP-2DcxSAJv	0.44	5.18	0.001	0.03	9.200	0.012	0.014	0.021	0.093	18.678	4.15	63.93	0.704	65.39	0.004	1.22
2015 TanSIS_TOP-2HEqMyTV	1.06	5.64	0.001	0.03	2.660	0.202	0.018	0.002	0.159	22.126	3.52	47.53	0.329	41.02	0.007	1.02
2015 TanSIS_TOP-3b8rVtmM	1.00	5.42	0.001	0.05	4.170	0.132	0.018	0.006	0.078	41.489	3.77	35.37	0.363	45.49	0.006	1.21
2015 TanSIS_TOP-3QZHqAOW	0.73	6.06	0.001	0.03	8.190	0.349	0.009	0.019	0.015	18.942	8.26	39.55	0.537	43.49	0.008	1.01
2015 TanSIS_TOP-3UtP80EJ	0.78	5.79	0.001	0.03	8.830	0.178	0.015	0.025	0.059	32.620	3.43	47.31	0.549	66.33	0.007	1.31
2015 TanSIS_TOP-4K3REmI6	1.32	5.64	0.001	0.04	3.990	0.049	0.011	0.002	0.067	39.636	4.52	53.83	0.403	50.23	0.110	1.42
2015 TanSIS_TOP-4W7JcoCa	0.77	5.93	0.001	0.03	6.420	0.437	0.019	0.019	0.097	38.240	6.10	33.36	0.574	56.98	0.010	1.33
2015 TanSIS_TOP-5LfrjyVD	0.75	5.53	0.001	0.03	10.09	0.019	0.015	0.003	0.937	35.868	3.66	40.16	0.779	65.55	0.005	1.32
2015 TanSIS_TOP-5ueT7Wyy	1.03	5.31	0.001	0.03	10.64 1	0.188	0.015	0.008	0.038	48.269	2.02	52.89	0.407	82.67	0.006	1.13
2015 TanSIS_TOP-69R5a2km	1.10	5.68	0.001	0.03	3.846	0.091	0.018	0.005	0.086	24.070	3.34	50.83	0.395	37.87	0.008	1.31

SSN	OC	pH	EC	TN	P	Ca	K	Mg	Na	S	Mn	Al	Zn	Fe	B	Yields
2015 TanSIS_TOP-6fSnteuo	0.92	5.92	0.001	0.04	6.680	0.395	0.025	0.030	0.182	35.272	6.55	45.87	0.486	53.74	0.007	1.21
2015 TanSIS_TOP-6PmIkGS7	1.17	5.47	0.001	0.03	12.14	0.442	0.016	0.013	0.039	23.823	3.70	52.91	0.578	49.49	0.015	1.01
2015 TanSIS_TOP-6ZYLwXVA	1.28	6.12	0.001	0.04	2.630	0.364	0.022	0.009	0.015	36.568	6.30	53.86	0.354	24.59	0.013	1.34
2015 TanSIS_TOP-7mHb8oLd	0.83	5.96	0.001	0.03	6.350	0.374	0.008	0.030	0.139	25.604	6.42	36.62	0.533	51.53	0.008	1.11
2015 TanSIS_TOP-7otmf2eR	0.91	5.77	0.001	0.04	5.980	0.195	0.024	0.018	0.025	47.467	3.58	27.00	0.415	63.57	0.006	1.11
2015 TanSIS_TOP-7rY0dTMC	0.97	5.91	0.001	0.02	6.480	0.045	0.014	0.020	0.031	44.260	2.33	42.14	0.521	52.54	0.007	1.32

The second was the 62 instances of Njombe randomly selected soil samples which were used to validate the model and this came from the Soil Care Depart of the Live Support Systems (T) LTD (LSSL) Soil Services Company. Figures 26, 27, and 28 show the respective laboratory-based off-the-shelf soil fertility Test results for the low, Adequate, and high fertility statuses, with the adequate level addressed as moderate soil fertility status.

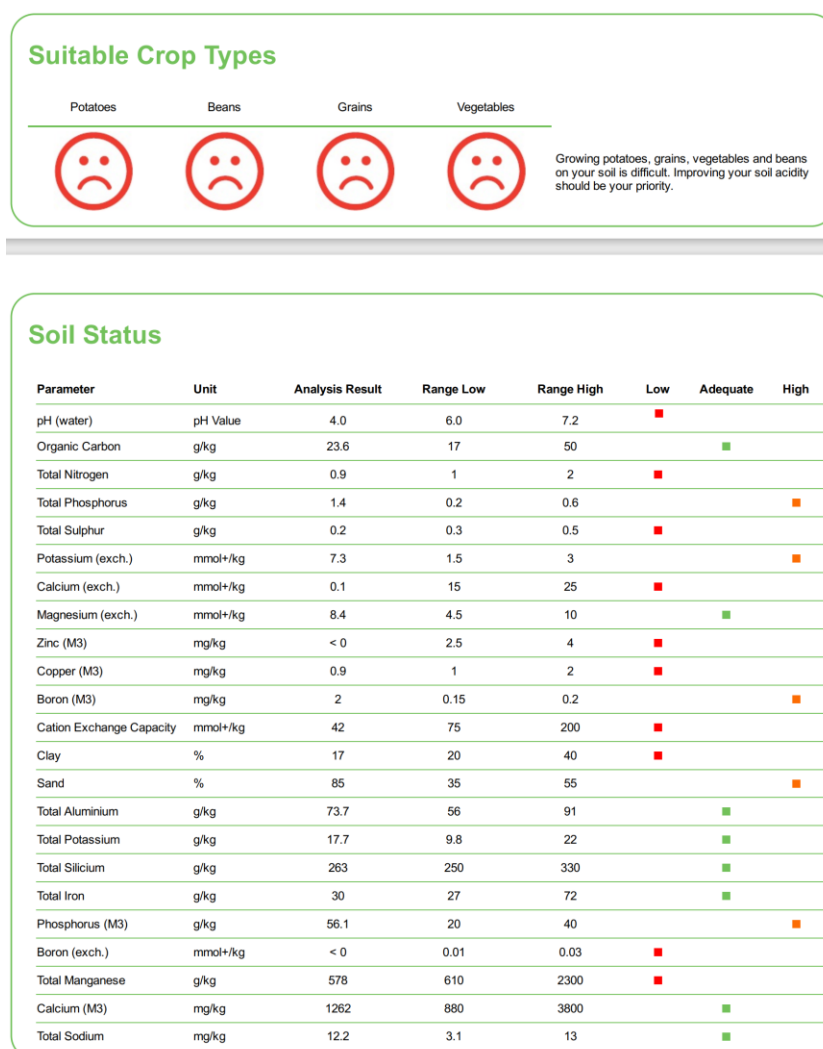


Figure 26: Laboratory-based off-the-shelf soil fertility test results – low

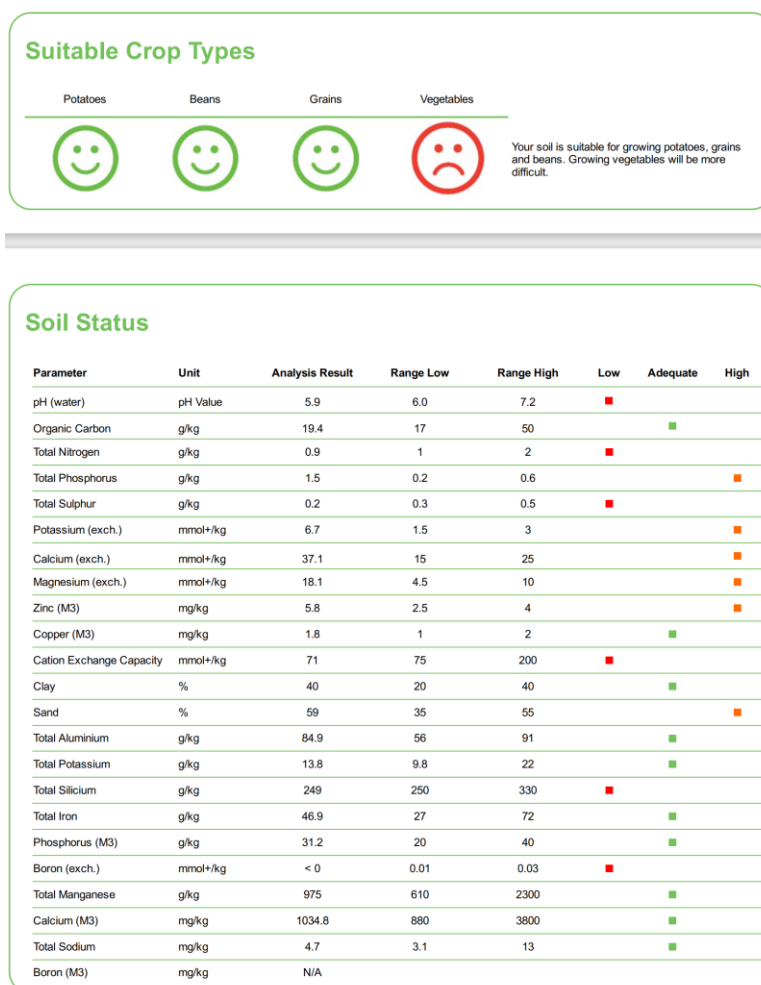


Figure 27: Laboratory-based off-the-shelf soil fertility test results – adequate

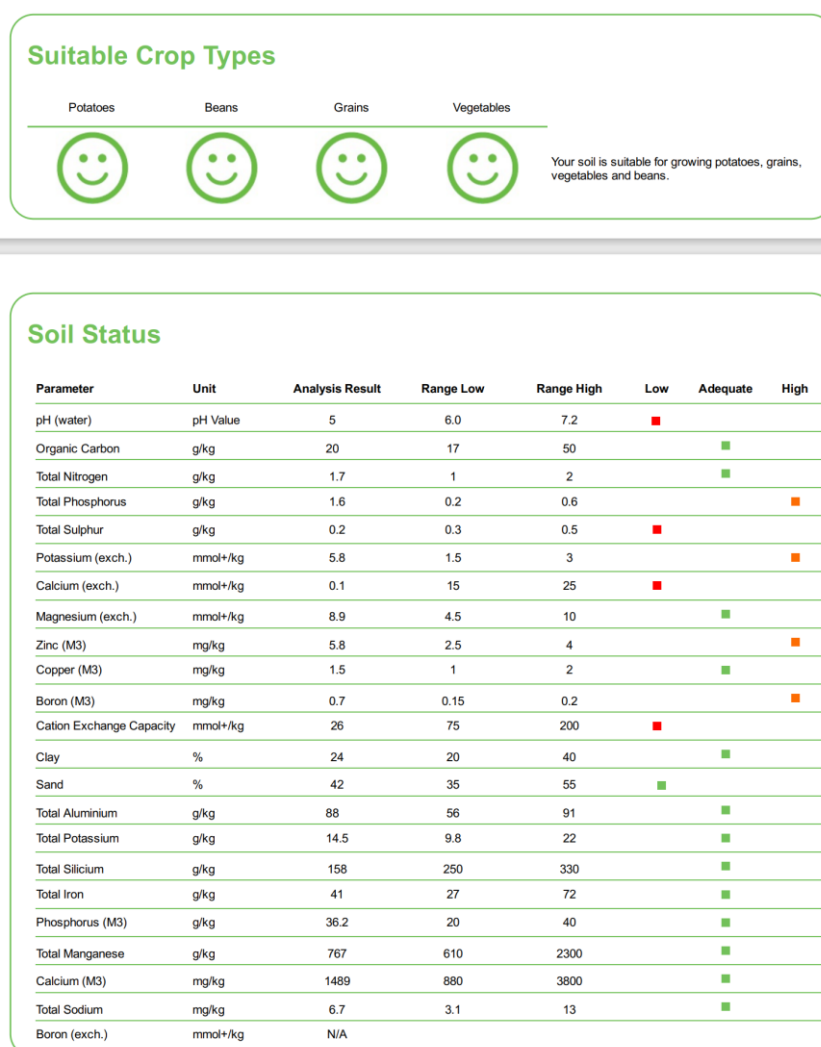


Figure 28: Laboratory-based off-the-shelf soil fertility test results – high

While a part of the Soil Care Laboratory-based Validation dataset is shown in Table 6. In addition, a triangulated experiment for evaluating the models' utility was done in Mbeya Uyole. Shown in Table 7 is the Model utility evaluation field experimentation soil properties collected data, whereby 64 soil samples were randomly selected with experimental field block-wise stratification in trials to ensure a fair representation of all model-based targeted field sections.

Table 6: Soil care laboratory-based validation dataset

Sample No	OC	pH	EC	P	Ca	K	Na	S	Mn	Al	Zn	Fe	B	Status
EAEON722A23	2.13	6.2	0.924	17	0.1	0.058	0.067	30	9.62	0.87	0.29	43.3	0.1	0
EAEON723A23	2.04	6	1.766	15	38.1	0.067	0.047	20	9.75	0.849	0.58	46.9	0.1	1
EAEON724A23	1.94	5.9	1.028	15	37.1	0.067	0.047	20	9.75	0.849	0.58	46.9	0.08	1
EAEON725A23	2.64	5.6	1.348	16	0.1	0.058	0.06	20	7.67	0.8	0.55	14.5	0.07	1
EAEON727A23	2.1	5.7	1.815	17	0.1	0.058	0.057	30	9.62	0.87	0.29	43.3	0.1	1
EAEON726A23	4.14	5.3	0.917	16	0.1	0.058	0.069	20	7.67	0.8	0.55	41	0.07	1
EAEON729A23	2.1	5.6	1.566	17	0.1	0.058	0.057	30	9.62	0.87	0.28	43.3	0.1	0
EAEON728A23	2.2	5	1.267	16	0.1	0.058	0.067	20	7.67	0.88	0.58	41	0.07	1
EAEON730A23	3.08	5	1.178	16	0.1	0.058	0.067	20	7.67	0.88	0.58	41	0.07	1
EAEON731A23	2.4	5.6	1.109	16	0.1	0.052	0.06	20	7.52	0.86	0.58	41	0.05	2
EAEON732A23	3.14	5.4	1.208	16	0.1	0.058	0.06	20	7.67	0.8	0.55	41	0.07	1
EAEON733A23	2.9	6.2	1.682	16	0.1	0.052	0.06	20	7.67	0.88	0.58	41	0.07	1
EAEON734A23	2.9	6.1	1.024	16	0.1	0.052	0.06	20	7.62	0.86	0.58	41	0.08	1
EAEON735A23	2.2	5.6	1.843	16	0.1	0.056	0.069	20	7.67	0.8	0.55	41	0.07	1
EAEON736A23	3.04	5.1	0.883	16	0.1	0.058	0.073	20	7.67	0.808	0.55	41	0.09	1

Table 7: Model utility evaluation field experimentation soil properties collected data

Sample No	OC	pH	EC	P	Ca	K	Na	S	Mn	Al	Zn	Fe	B
S1	1.001	5.817	1.562	10.115	0.156	0.017	0.013	47.31	3.793	0.642	0.52	40.073	0.009
S2	2.217	5.038	1.295	4.571	2.565	0.05	0.084	51.66	6.732	0.502	0.717	34.624	0.009
S3	2.424	6	1.807	8.357	3.86	0.08	0.452	64.513	5.234	0.896	0.801	47.657	0.021
S4	1.073	5.906	0.548	7.22	0.283	0.018	0.161	30.109	4.571	0.453	0.477	60.513	0.015
S5	0.772	6.139	0.932	14.314	0.263	0.047	0.013	44.251	9.558	0.375	0.998	44.12	0.017
S6	1.217	6.216	1.134	11.276	0.398	0.015	0.021	57.007	10.734	0.376	0.831	46.988	0.024
S7	0.455	6.174	1.629	15.157	0.515	0.015	0.048	43.607	12.908	0.328	1.156	58.361	0.029
S8	1.065	6.083	1.014	11.024	0.174	0.02	0.183	29.281	6.255	0.303	0.873	41.417	0.018
S9	1.02	6.019	1.656	15.426	0.212	0.025	0.093	38.863	3.652	0.419	0.518	39.496	0.013
S9	1.338	6.199	1.406	8.975	0.302	0.018	0.016	76.509	7.647	0.347	0.871	38.957	0.013
S10	1.055	5.545	0.697	7.526	0.218	0.019	0.021	30.07	4.422	0.547	0.482	46.96	0.016
S11	1.296	5.891	1.177	7.945	0.183	0.017	0.22	45.947	10.693	0.352	0.497	33.353	0.019
S12	1.019	6.386	1.123	7.524	0.776	0.017	0.19	58.548	5.749	0.432	0.583	45.547	0.014
S13	0.978	6.248	1.924	14.611	0.455	0.017	0.182	57.452	9.762	0.275	0.85	35.326	0.017

3.3.3 Heterogeneous Hybrid's Weighted Voting Ensemble Experiment Setups

A hybrid of both Unsupervised and Supervised machine learning algorithms was used to implement the required 2-Staged heterogeneous hybrid ensemble committee (2S-HHEC) machine for improving soil fertility status predictive performance experiments, as shown in Fig. 29 of the 2S-HHEC experimental setup which created the hybrid classifiers following implementation of unsupervised learning on the soil data, evaluation of individual ML algorithms classifiers to the development of optimized WVE for reliably predicting soil fertility status prediction at improved performance through the execution of the proposed “1EXP(-)Z⁺ initial term based arithmetic sequences multi-precision search spaces algorithm function for systematic brute exhaustive optimization of intelligent small WVE (1EXP (-) Z⁺_{IT}-ASMPSS-BEO-_{IS}WVE)”.

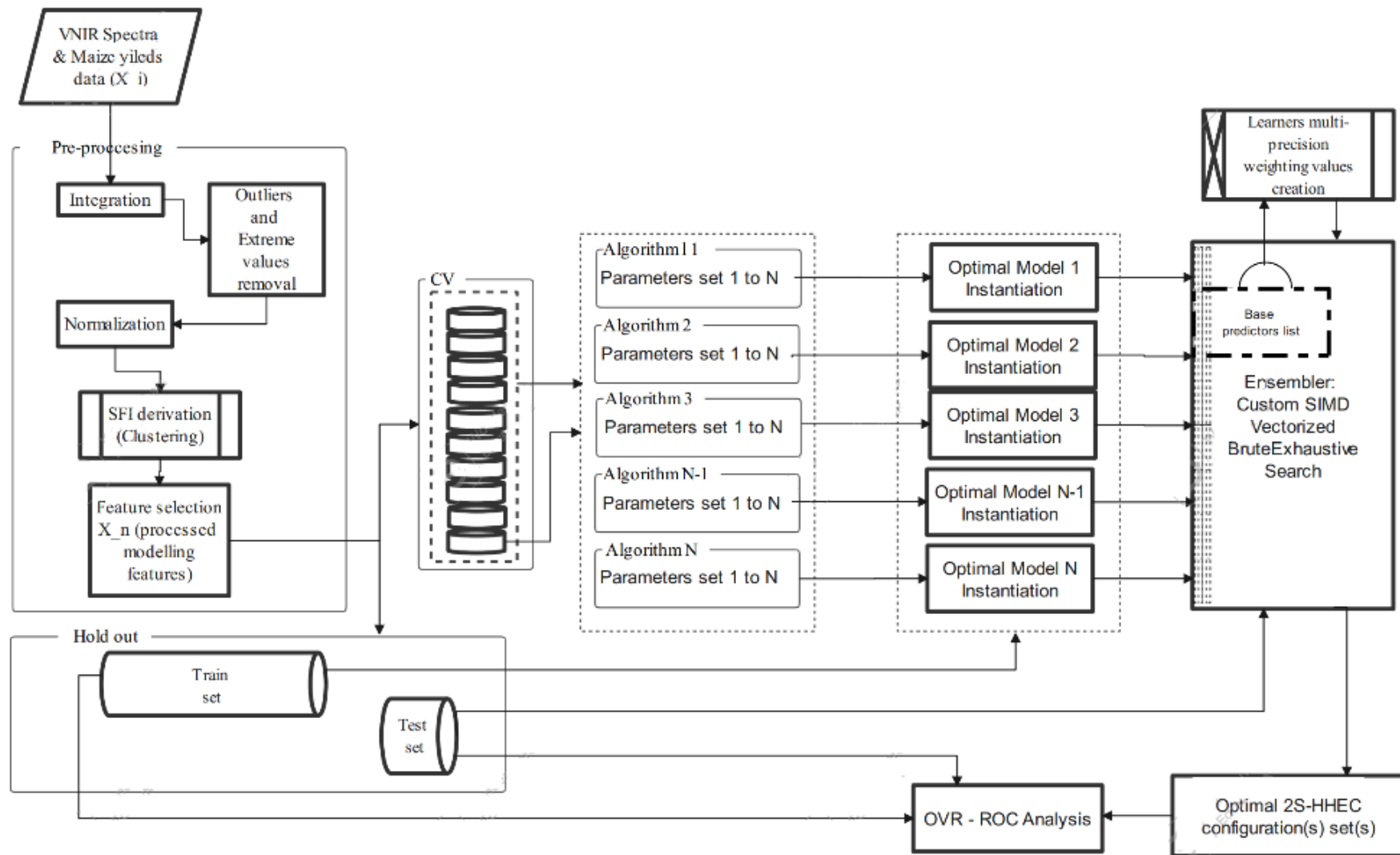


Figure 29: The 2S-HHEC Experimental setup

(i) Data preprocessing

The collected data was pre-processed by separating the top (0-20 cm) instances from those of the subsoil (20-50 cm), and duplicates were removed. Soil properties and maize yield data were aggregated to form the set before the required machine learning problem modeling dataset.

(ii) Dataset quality testing

The quality of data is one important aspect for data scientists and statisticians, whereby they would aim to understand the distribution(s) present in the data to be able to apply appropriate measures and procedures for better interpretation of the results (Varshney, 2020). Whereas, the Shapiro Wilk normality test is one of the data normality test techniques (Malato, 2022; Royston, 1983; Royston, 1992; Yazici & Yolacan, 2007), herein we employed the quantile-quantile (QQ) or simply quantile plots which aid in the visualization of the distributions available in the random variables by plotting these random variables on the y-axis, and the normal distribution on the x-axis, such that the plot between would a visualization of the present data distribution such that if the quantile points lies across the straight line $y=x$ then it is a normal distribution, otherwise if the right side is above the $y-x$ line and the left side is around the line, then it is right-skewed, likewise if the right side is around the line and the left is below then it is a left-skewed (Chan, 2022; Larasati *et al.*, 2019; Varshney, 2020). This determination of which will aid in the requirement for the application of data normalization procedure before the effective application of consequent analytical and modeling techniques which work best at Gaussian distributions, and resultant models calibration, otherwise remedies such as data stratified sampling techniques could only aid if the issue was an imbalance type of concern.

Therefore to address the quality of our data, the Sci-kit learns scipy module's skew method was run on boxplots to create the Q-Q plots to visualize data abnormality before final normalization into more ML tractable modeling data (Chan, 2022; PyShark, 2021; Turing, 2023), whereby some variables exhibited outliers and extreme values which were later on removed, by using the Interquartile Range statistical measure method for noise filtering and reduction, before normalization of the random variables which could that were transformable to reduce randomness were possible.

(iii) Fertility Index Derivation and features selection

The soil data used in this research initially contained no distinct class labels. In the first stage of the 2-stage hybrid implementation, an additional feature, i.e. the fertility class label necessary for use with the features in the second ML algorithm-based classification stage of the 2-stage hybrid was created. This was accomplished by applying an unsupervised machine learning K-means algorithm that is readily available in the sci-kit learn library which is contributed by Pedregosa *et al.* (2011) and the automatic knee detection method (K-Elbow) to model for the optimal number of fertility classes as characterized in the data, analysis of variance (ANOVA) test was performed on the formed groups to test for groups similarities otherwise differences, with a null hypothesis that “the groups are different”, this was accomplished by Tukey honest significant difference (HSD) test. Such clustering method is common for such a cluster grouping task and has been used in many studies in various other domains, whereby a review of the use of the approach to characterize data into common groups could be found (Nyambo *et al.*, 2019), whereby the use of the approach to describe smallholder farmers into groups with similar characteristics were highlighted. It is necessary to avoid model performance impairment due to complexities that may be caused in many cases by extensive multicollinearity in data (Alin, 2010). Thus removal of highly correlated dataset feature(s) is crucial to eradicating unnecessary multicollinearities in data. For that purpose, we performed feature selection by identifying and removing features with a correlation above 70%, as a standard correlation threshold in most studies. Automated derivation of the class label feature is not a new practice, given the challenges of processing large amounts of data to obtain such information using expert knowledge alone, it made automatic derivation becomes imperative to effectively derive target classes.

(iv) Model Selection

In this research a total of seven ML classifiers were evaluated by using the Sci-kit learn powerfully ML libray by developed Pedregosa *et al.*, (2011), These are support vector machine (SVM), DecisionTreeClassifier(DT), GaussianNB (NB), K-Nearest Neighbors (KNN), AdaBoost (AdaBoost), Gradient Boosting (GB), and Random Forest (RF), of which were used with Stratified KFold Cross Validation (N=10) on the soil dataset to robustly select best models across a wide range of their associated tuning parameters.

While the clustering algorithm that was used to model the new fertility index feature could have been implemented with an integrated custom distance-based prediction module, In this research an evaluation of the stated already existing very advanced and powerful classical machine learning classification algorithms were opted due to:

- (i) The key aim of the algorithms' existence is specifically to implement models for the problem task in hand, that is, the development of classifiers for predicting soil fertility status.
- (ii) Profound potentials have already been theoretically and empirically demonstrated performances in various ML classification problems.
- (iii) The abilities of the algorithms to handle complex data, and to scale into different domain-specific classification tasks, as observed from the literature whereby these have been applied to solve various problems, including health and medical applications, banking and finance, network securities, transportation, and agriculture.

Their ability to increase in understanding during training as the associated learning data increases makes them the first option in research and development endeavors that involves large amounts of data as their key requirement for successfully data-driven predictive solutions to be probably developed.

3.3.4 Base Models Performance Improvement Through WVE

After creating and evaluating the performances of individual base models, the performance improvement procedure was implemented through the development of various WVE base models class probabilities prediction combinations by applying the soft voting method and selection of best ensemble weights coefficients values. These which was achieved by using brute exhaustive search procedures that incorporate the proposed novel 1EXP (-) Z^+ search space initialization function for the provision of the possible weighting values necessary to implement an optimization process in search of the required optimal configuration sets of base models and corresponding weights, specifically at variable search space precision values to attempt the optimization even at much bigger precisions.

3.3.5 Development of the High Performance put Brute Exhaustive WVE 1EXP (-) Z+ Optimization Algorithm

Lemma 1

From Equation (1) in Subsection 2.1.7. of the weighted voting ensemble scheme for model performance improvement, If the WVE combination equation (1) that is described by Escorcia-Gutierrez *et al.* (2022), when expressed as in equation (3) of its matrix form Y , that expresses a mathematical system of linear equation's that can be operated through matrix operations to compute the overall prediction outcomes for each WVE's combinations as a summation of the product of weights coefficients W_i and j base experts class probability predictions C_1 to C_j on dataset D having d unseen targets instances values, where $i > 1$, and $j > 1$.

$$Y = \begin{bmatrix} W_1C_1 & W_1C_2 & W_1C_3 & W_1C_j \\ W_2C_1 & W_2C_2 & W_2C_3 & W_2C_j \\ W_3C_1 & W_3C_2 & W_3C_3 & W_3C_j \\ \vdots & \vdots & \vdots & \vdots \\ W_iC_1 & W_iC_2 & W_iC_3 & W_iC_j \end{bmatrix}, \quad (3)$$

Thereby, Y can be compared against true classes to score the prediction accuracy of the WVE, which for all other possibly available WVE combinations, the optimal set is chosen based on the one which satisfies an established criterion such as error minimization, accuracy, or other performance measure maximization as an objective function.

Proof

The above Lemma has been noted in Escorcia-Gutierrez *et al.* (2022). Whereby the values of the weights coefficients were referenced as a function of the individual WVE base learners' $f1_score$ performances for evaluating the efficiency of individual learners in the ensemble during training.

Whereby using equation (1) of the weighted voting ensemble scheme, in subsection 2.1.7., the WVE can be represented as a system of linear equations (Pospíšil, 2020; Wedderburn, 1915),

$$Y_1 = W_1C_1 + W_1C_2 + W_1C_3 + \dots + W_1C_j$$

$$Y_2 = W_2C_1 + W_2C_2 + W_2C_3 + \dots + W_2C_j$$

$$Y_3 = W_2C_1 + W_3C_2 + W_3C_3 + + W_2C_j$$

.

.

.

$$Y_k = W_iC_1 + W_iC_2 + W_iC_3 + + W_iC_j$$

These of which can be in matrix form as shown in equation (4)

$$Y[k] = \begin{bmatrix} W_1 & W_1 & W_1 & W_i \\ W_1 & W_2 & W_2 & W_i \\ W_2 & W_3 & W_3 & W_2 \\ \vdots & \vdots & \vdots & \vdots \\ W_i & W_i & W_i & W_i \end{bmatrix} * \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_j \end{bmatrix}, \quad (4)$$

(i) The Multi Precision Search Spaces Formulation Function

In general, the generation of a WVE of ML classifiers may consider mostly two phases that are: a) Using various candidate ML algorithms to generate potential base members' classifiers that are to be used to form the WVE combinations, and b) selection of base models' optimal weights based on the WVE combination grounded by an accuracy performance criteria.

Proposition 1

If instead an ordered weights coefficients matrix $W[k][n]$ can be automatically generated from the permutation of an explicit vector $W[n]$ *that is* referred to as the search spaces Sp and Sp_{z+} *herein*, of weight values that satisfy the WVE weights coefficients domain constraints in equation (2), with a variable matrix $C[j][d]$ of j base expert's class probability predictions on dataset D *containing* d total instances. Such that the resultant WVE combination k constant predictions matrix $Y[k][d]$ *or* Y_{pred} , can be obtained from the product of the ordered weights coefficients matrix $W[k][n]$ and variable matrix $C[j][d]$ as shown in (5), which is augmented from equation (4) with the appending of the dimension of the dataset instances I_d , for practical optimization purposes.

$$Y[k][d] = \begin{bmatrix} K1W1 & \cdots & K1Wn \\ \vdots & \ddots & \vdots \\ KkWn & \cdots & KkW1 \end{bmatrix} * \begin{bmatrix} C1I1 & \cdots & C1Id \\ \vdots & \ddots & \vdots \\ CjI1 & \cdots & CjId \end{bmatrix}, \quad (5)$$

At that juncture, assuming that the *variable* matrix $C[j][d]$ of m classifiers class probability predictions on instances I of dataset D with length d are provided, and an initialization function for explicit formulation of values for generating the weight coefficients matrix $W[k][n]$ which satisfy WVE weights constraints in equation (2) can be derived and developed as part of an automatic weighting values generation algorithm, then a Brute-exhaustive optimization procedure can be applied to search one optimal combination set from the automatically created WVE combinations predictions matrix in equation (5). This whose general form is that in equation (3). Whereas, equation (4) serves to compute the general form in equation (3) as a product the of the weight coefficients $W[k][n]$ and variable matrices $C[j][d]$ of j individual classifiers probability predictions on supplied dataset d as represented in equation (5). But rather this time, the weight coefficients are automatically generated, hence the complete WVE general form in equation (3) will be automatically generated. It is to be proved that the general WVE combination matrix form representation in equation (3) can be automatically generated. As such, specifically for practical optimization purposes, the brute exhaustive search can be automatically applied as long as dataset D with instances exists.

Proof

First, the variable K which represents the combinations counts is introduced into equation (3) to obtain a new representation form as in (6),

$$Y[k] = \begin{bmatrix} K1W1C11 & \cdots & K1WnCj \\ \vdots & \ddots & \vdots \\ KkWnC1 & \cdots & KkWnCj \end{bmatrix}, \quad (6)$$

Whereby K keeps track of the formulated WVE combination prediction. Then a function is derived to initialize the weights variable values, and incorporated in an algorithm that applies the arithmetic sequences to formulate the other values. Further derivation and complete weight values generation algorithm are explained starting with the operationalization architecture of the algorithm function shown in Fig. 30 of the architectural operationalization of the proposed algorithm core function for the formulation of respective first terms an arithmetic sequence and generation of the prospective sequence values. Whereby the complete operationalization illuminates as follows, a search space referenced by a positive integer denoted as Z^+ , is initialized to 1 representing search space 1, and then is used to computationally generate the corresponding sequence's first term from which all other elements a_0 to a_n of the sequence can be computed to formulate a particular sequence these which will define the actual weights

values w_1 to w_n specifying our WVE domain, of which the algorithm later process these values by permutation with WVE base models to generate various WVE subsets combinations referred as search space which are used as part of the algorithm brute exhaustive based procedure implementation proposed in proposition 1.

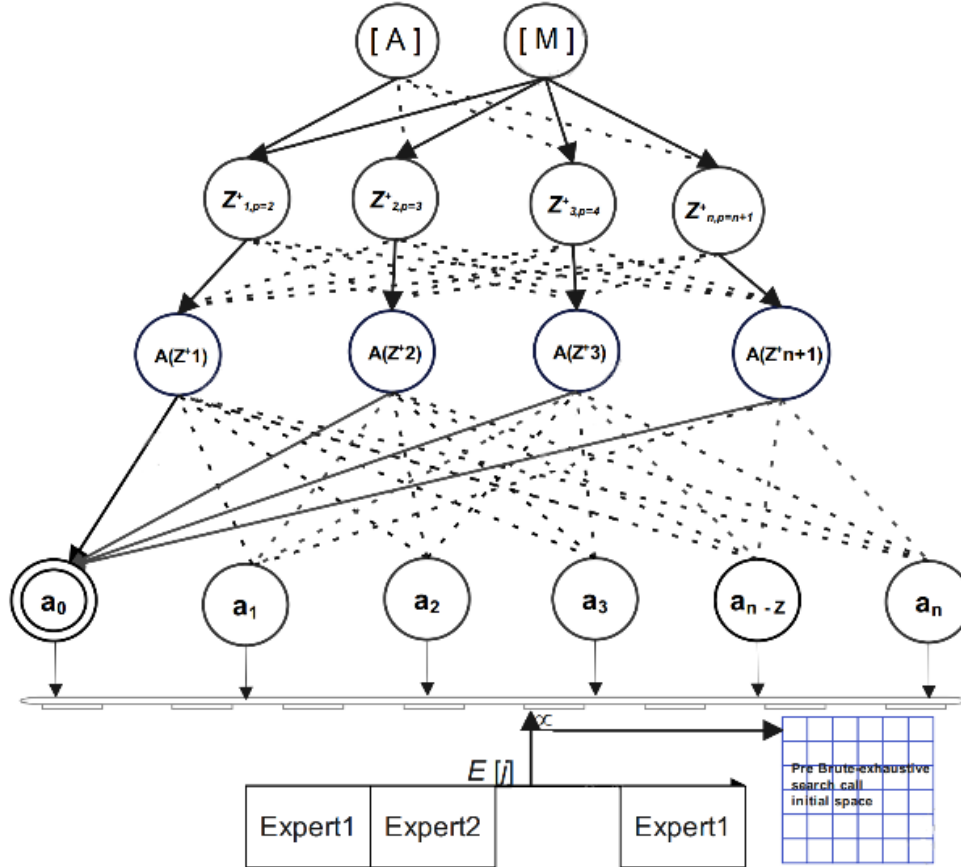


Figure 30: The proposed multi-precision weights formulation operational architecture

(ii) The 1EXP (-) Z^+ initial term based Weight Coefficients Values Formulation Function

To derive the required function, its closed-loop equation was formulated. First, we derive the function for formulating the stated multi-precision arithmetic sequences as input arguments to the search space generation procedure. Whereas, the Taylors series can often be used for the derivation of an algorithmic system's closed-loop equation that expresses a particular problem domain. Through lemma 1 and proposition 1, herein we used the arithmetic sequence to substantiate the proposed 1EXP (-) Z^+ _{IT}-ASMPSS-BEO-_{IS}WVE algorithm, whereby the sequence's first-term a_0 serves as a principle for generating the other respective a_1 to a_n term

of values of the arithmetic sequence A in equation (7) by using the arithmetic sequence's closed-loop equation (8),

$$A = \sum_{n=1}^{(1/a_0)} (a_0 + a_0 * n), \quad (7)$$

$$a_n = a_0 + d * n, \quad (8)$$

whereby a_0 is the first term of the sequence, which is initialized by the proposed 1EXP (-) Z^+ initial term-based initialization function $F(Z^+)$ in Equation (9) as floating point numbers (FPN) due to apprehending the constraints in Equation (2), whose computational notation is shown in Equation (10), to form a 1EXP (-) Z^+ initial term based FPN weight value n arithmetic sequence A_{Z^+} whose first term is initialized in Equation (9), and these are substituted in Equation (7) to reflect the proposed arithmetic sequence A_{Z^+} formation expression in (11), this which forms the basis of the weights coefficients matrix values.

$$F(Z^+) = 1\text{EXP} (-) Z^+, \text{ for } Z^+ > = 1, \quad (9)$$

$$F(Z^+) = 1e^{-Z^+}, \text{ or simply } 1/(1eZ^+) \quad (10)$$

$$A_{Z^+} = \sum_{n=1}^{(1/1e^{-Z^+})} (1e - Z + 1e - Z * n), \quad (11)$$

where Z^+ denotes all positive integers greater than zero and less or equal to the reciprocal of one exponent negative Z^+ , that is, $1/(1\text{EXP} (-) Z^+)$.

(iii) The 1EXP (-) Z^+ based Weights Coefficients Matrix and Search Spaces Matrix Computation

The search space matrix SP_{Z^+} of K combinations can then be generated as a permutation of sequences A_{Z^+} and base expert's list vector $C[j]$, as represented in equation (12).

$$SP_{Z^+} = \text{permutation} (A_{Z^+}, C[j]), \quad (12)$$

Based on the $1e^{-Z^+}$ initialized values sequences A_{Z^+} in equation (11) substitution in (12), we obtain equation (13), representing the $1e^{-Z^+}$ based spaces Spz^+ required for the coefficient and variable matrices distillation.

$$Spz^+ = \text{permutation} ((\sum_{n=0}^{(L)} (1e - Z + 1e - Z * n)), C[J]), \quad (13)$$

Where L is the reciprocal of the initialized fractional value based on $1eZ^+$, and Z^+ is greater or equal to 0. A re-arrangement of the generated permutation spaces Spz^+ , from equation (13) in order of the dimensions of the variable matrix representing available classifiers class probability predictions would represent weighted output predictions for K combinations as expressed in equation (14).

$$\begin{bmatrix} Y1K1 \\ \vdots \\ YkKk \end{bmatrix} = \begin{bmatrix} K11\exp - zC1 & \cdots & K11\exp - z + 1e - Z * n Cj \\ \vdots & \ddots & \vdots \\ Kk1\exp - z + 1e - Z * n C1 & \cdots & Kk1\exp - zC(n^j) \end{bmatrix}, \quad (14)$$

By decomposing the matrix in equation (14) into its constant, coefficients, and variable matrices as explained in (Bellman, 1997; Kittappa, 1993), we obtain equation (15), which computes the constant matrix as an output prediction as a product of the coefficients, variable matrices of the general WVE matrix form in equation (14).

$$\begin{bmatrix} Y1K1 \\ \vdots \\ YkKk \end{bmatrix} = \begin{bmatrix} K11\exp - z & \cdots & K11\exp - z + 1e - Z * n \\ \vdots & \ddots & \vdots \\ Kk1\exp - z + 1e - Z * n & \cdots & KkK11\exp - z \end{bmatrix} * \begin{bmatrix} C1 \\ \vdots \\ Cj \end{bmatrix}, \quad (15)$$

This which when subjected to class probability predictions on the dataset with d instances, could also be represented as

$$\begin{bmatrix} Y1K1 \\ \vdots \\ YkKk \end{bmatrix} = \begin{bmatrix} K11\exp - z & \cdots & K11\exp - z + 1e - Z * n \\ \vdots & \ddots & \vdots \\ Kk1\exp - z + 1e - Z * n & \cdots & KkK11\exp - z \end{bmatrix} \begin{bmatrix} C1I1 & \cdots & C1Id \\ \vdots & \ddots & \vdots \\ CjI1 & \cdots & CjId \end{bmatrix},$$

Finally, the vector Y or $Y[k][d]$ of equation (15) will be calculated as the argument max of the product of weights coefficient matrix and classifiers class probability predictions, which is then scored for accuracy against the true targets as observed in the data set D with I instances, for each k combination the accuracy is compared with the previous maximum score to pick it as a new maxim if the previous is small otherwise the algorithm proceeds to the next combination iteration k . Until terminations conditions, the k th combination with maximum accuracy is returned as the optimal WVE combination configuration set.

Whereas the final automatically generated search combinations in equation (15) are similar to the general WVE matrix equation (4) which was decomposed from matrix (3) in Lemma 1. Equation (14) and its decomposed form in Equation (14) are also similar to the WVE matrix

forms in Equations (5) and (6) in the initial proposition 1. As the automatically generated combinations represent a system of linear equations through the presented matrices, these of which have been proven to suffice for WVE predictions computations in equation (1). It entails that the automatically derived matrix form based on our proposed arithmetic sequences weights coefficients formulation function can well serve for representation of K possible WVE combinations in equation (1). Hence it has been proved that the general WVE combination matrix form representation in equation (3) can be automatically generated. These which can then serve as automatic synthetic search space for brute exhaustive search could be implemented.

(iv) The Complete 1EXP (-) Z+IT-ASMPSS-BEO-ISWVE Algorithm

As a modification of the straightforward implementation brute exhaustive search algorithm 3. As shown in Fig. 31 of the proposed algorithm's flowchart, the corresponding pseudo-code shown in algorithm 3 of the proposed 1EXP (-) Z^+ initial term-based sequences formulation and weights coefficients matrix generation algorithm was formulated to present its computational procedures before actual machine implementation for brute exhaustively optimizing WVE(s).

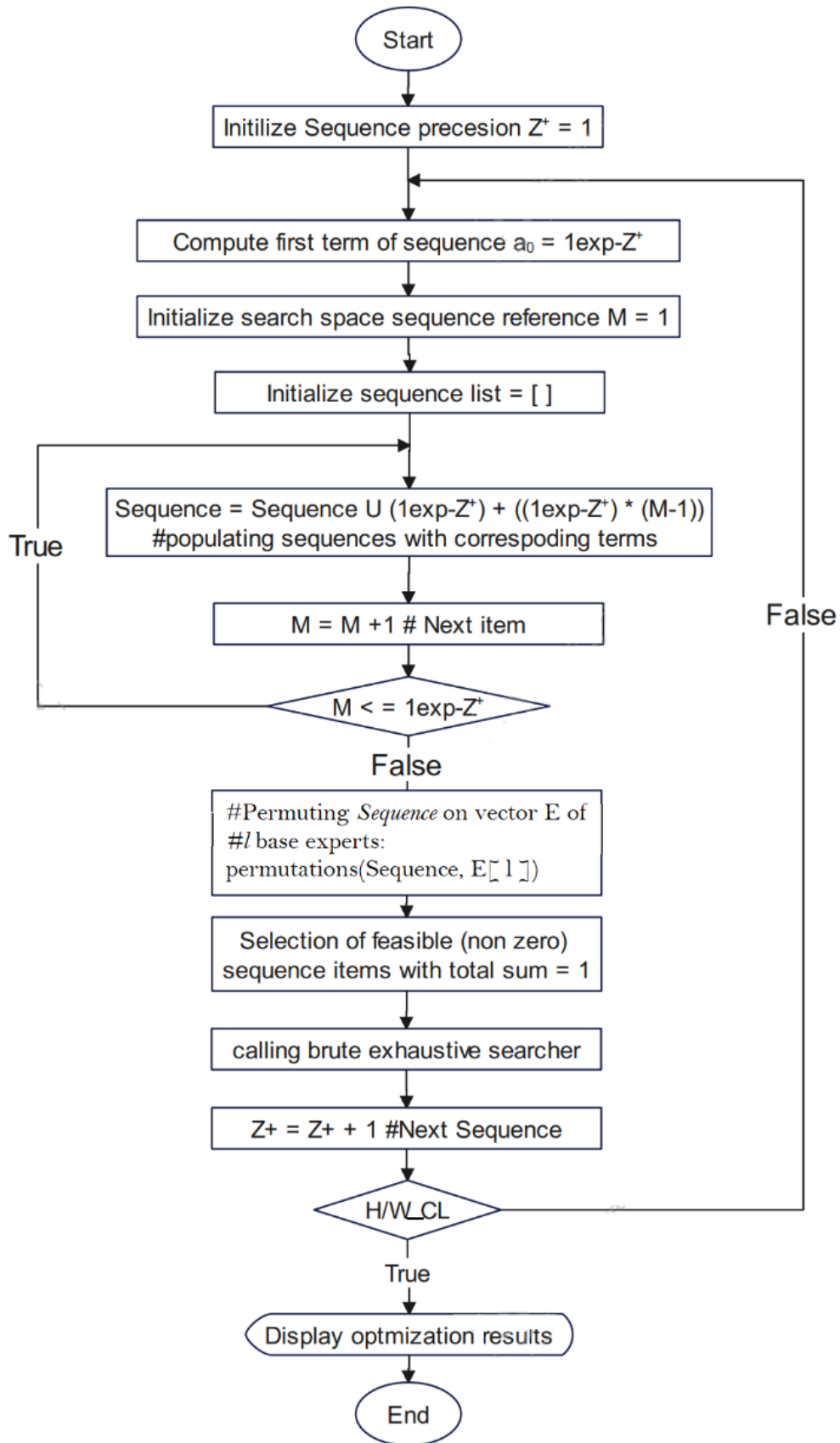


Figure 31: The full 1EXP (-) Z+IT-ASMPSSA_BES_ISWVEO flowchart

Algorithm 3. 1EXP (-) Z^{+IT} -ASMPSS-BEO- $ISWVE$ Algorithm Pseudocode

Input: Base experts' Probability predictions, true targets

1. Start
2. initialize search space precision (Z) = 1
3. **REPEAT**
4. Compute the first term of the sequence as $a_0 = 1e - Z$
5. Initialize Search_space reference $N = 1$
6. **REPEAT**
7. Compute nth term a_n , $a_n = (1e - Z) + ((1e - Z) * N)$
8. Sequence = \cup .Sequence + a_n
9. Increment $N = N + 1$
10. **UNTIL** $N \leq 1eZ$
11. $SP_{Z+} = \text{permutations}(\text{Sequence}, E[j])$
12. Brute_Exhaustive_optimization(SP_{Z+} , $C[j][d]$)
13. Increment $Z = Z + 1$
14. **UNTIL** Z reaches computational lim. or combination k
15. Display optimization results
16. End

Output: high-performance WVE subsets weight estimates

The proposed algorithm generates the weights coefficient values through their formulations from arithmetic sequences-based function procedure, where it later invokes the brute exhaustive searching procedure with a procedural call as part of an integrative implementation thereof. Mainly the algorithm initializes the sequence reference as shown in step 3 after the start of its execution. Then until step 11 the search spaces SP_{Z+} are generated, with an additional built-in procedure to re-ensure the formed SP_Z still satisfies the condition $\sum_{i=1}^n Wi = 1$, as such it should be considered as a potential solutions pool. In step 12 the brute exhaustive procedure is called to search the formed search space SP_{Z+} and return an optimal weights configuration set from the corresponding SP_{Z+} based on class probability predictions $C[j][d]$. In step 13, the next sequence is initialized. In step 14, the algorithm checks if objective criteria and computational capacity are still not limited the process repeats until either one or both of the termination conditions are satisfied. Finally, it provides the optimization results in step 15 before ending the execution in step 16.

(v) **The 1EXP(-)Z+ initial-term based WVE brute exhaustive optimization package experimental setup**

Finally, the complete package of 1EXP (-) Z⁺ initial-term based arithmetic sequences multi-precision search spaces for brute exhaustive optimization of intelligent small WVE algorithm procedures codes were implemented by using Python using the Scikit-learn machine learning library functions (Pedregosa *et al.*, 2011). Contrary to using the traditional single instruction single data (SISD) computational operations implementation, vectorization array programming for single instruction multiple data (SIMD) operations which provide fast computations (Raskulinec & Fiksman, 2015) was necessary given the nature of the custom integrated much finer weights values formulation module for automatic multi-precision which may require many computations. Hence they are suitable for computations that involve extensive iterations, for that reason, SIMD was used to run the search on automatically unveiled search grids. Fig. 32 shows the higher abstraction of the proposed algorithm package for searching appropriate WVE's base models weights configuration set.

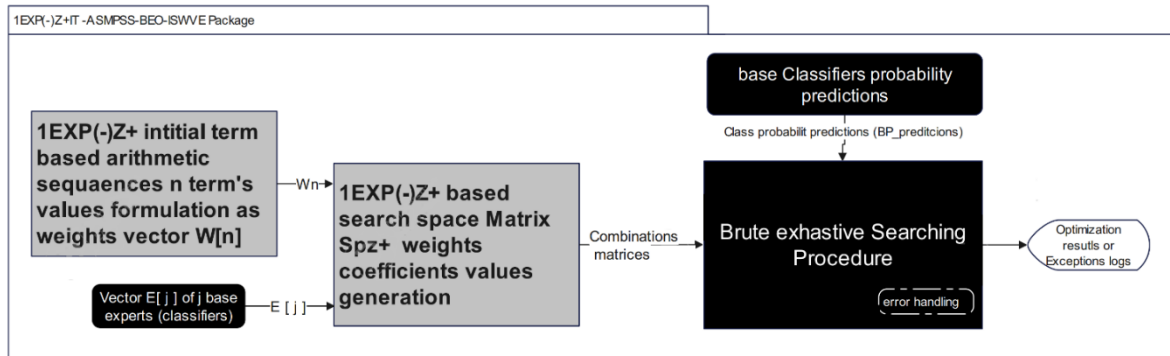


Figure 32: The 1EXP (-) Z⁺IT -ASMPSS-BEO-ISWVE package diagram

This was executed using an experimentation setup shown in Fig. 33, to evaluate its efficiency in formulating search spaces as well as its effectiveness in estimating optimal WVE across the various formulated systematic weights combination search spaces. Whereby the base experts' class probabilities predictions BP_Predictions resulting from an evaluation of dataset Instance's features are scored against true targets for accuracy by using Python's sci-kit-learn scoring library function to determine the accuracy of the combination as an objective criterion. Eventually, the selection of the WVE combinations output predictions with maximum accuracy could be determined from the run across the entire system of K combinations.

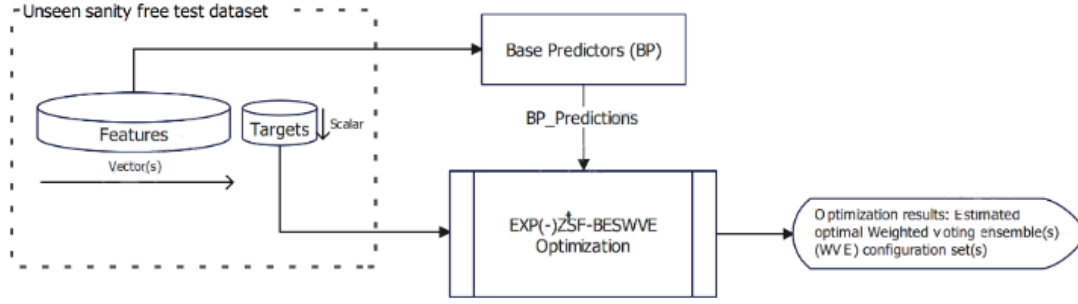


Figure 33: The 1EXP (-) Z+IT-ASMPSS-BEO-ISWVE experimental setup

3.3.6 Development Environment

This research model and the proposed optimization algorithm procedures code developments were done using Python programming language because this language provides vast support of libraries functions for machine learning modeling (Pedregosa *et al.*, 2011). The implementations were executed on Intel(R) Core(TM) i7-8550U CPU @ 1.99 GHz with 16 GB RAM, as well as in the Core i8 hardware with 64 GB RAM, 64-bit operating system, which produced a result set constituting of similar results from Core i7, with more additional results due the Core i8 hardware capacity which permitted for more computations.

3.3.7 Mathematical Co-processor Computational Limitations

However, whereas on one hand, the establishment of systematic search spaces formulation computations for brute exhaustive search procedure implementation is an effort of utility to WVE optimization on computer hardware, according to the no free lunch theorem, the formulated values which are floating point numbers (FPN) would likely face the FPN math co-processor computational or memory allocation in support of the defined by the corresponding floating-point arithmetic (FPA) adder, subtract, multiplier, and divider operations computational hardware architecture's memory allocation for the sign, exponent, and fraction also termed as mantissa memory allocation limitations as defined by the IEEE 754 standard for FPN arithmetic operations permissible operations and exceptions specification in Committee (2019). Particularly in our case that may prevail in the case of computing very high-scaled precision fractional FPN weighting values, entailing the prevalence of possibilities for increased combinations to the limitation of the available hardware memory capacity.

Shown in Fig. 34 is a microprocessor FPU memory allocation, whereas the FPN sign bit of the algorithm's formulated weight values adhere to the constraints in equation (2), which makes them always positively signed, the apprehension stems much in the exponent and fraction part of the FPN value, as the exponents part of a very large FPN value may shift into the mantissa segment of the memory registers locations hence processed as part of the fraction also known as mantissa, in turn leading to nonsensical in the best case, while an overflow is an expectedly worst case scenario.

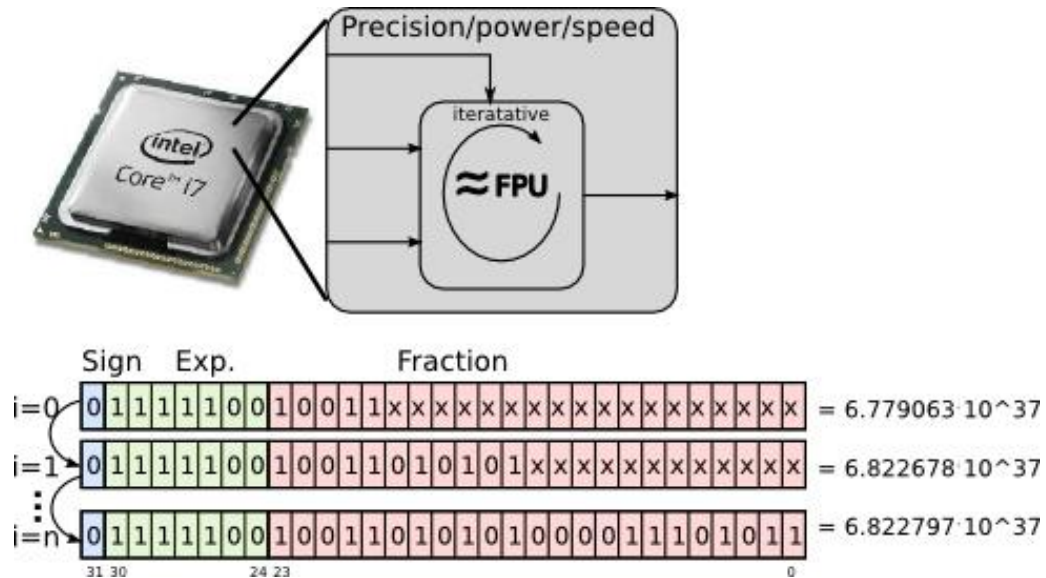


Figure 34: Micro-processor FPU

The stated computational memory limitation requires proper handling otherwise it may result in nonsensical or in worst-case scenarios overflows of the memory buffer if underlying hardware capacity is not supportive for search operations on the large unveiling search spaces of candidate solutions combinations. Unlike, the classical bit representation with only two states 0 and 1 in the FPNs word sizes memory architecture that may raise FPN memory limitations. On the contrary, the quantum qubit variant may become among the key considerations as it provides for much more storage states other than just 0 and 1, which are represented by an arrow pointing to a qubit spherical formation.

Figure 35 illustrates the classical bit qubit representation for quantum computations. Technically, qubit mechanics are based on a probabilistic measure to store and extract information that was previously stored in the qubit states, where the arrow will point where the information should be extracted, north if the state is 1, south if it is 0, otherwise to superimpose it as a 0 or 1 based on the other locations of the quantum spherical formation, in turn, it encodes

an infinite sequence of digital information which must later be extracted by a measurement that will result into an ordinary bit information of 0 or 1. Whereby quantum qubit combinations would possibly allow for the formulation of very big qubit information structures for storage and retrieval of enormous computational variable values to be processed in a quantum machine, hence overcoming the pre-stated memory allocation limitation of the classical bit system, but that would mean to shift the entire algorithm computational package into the quantum format. This will also lead to a reduction in energy costs as quantum computers run within minutes a task that these other non-quantum computational environments can take even up to a month. As such it is a green computing potential candidate concerning environmental preservation (Jaschke & Montangero, 2023).

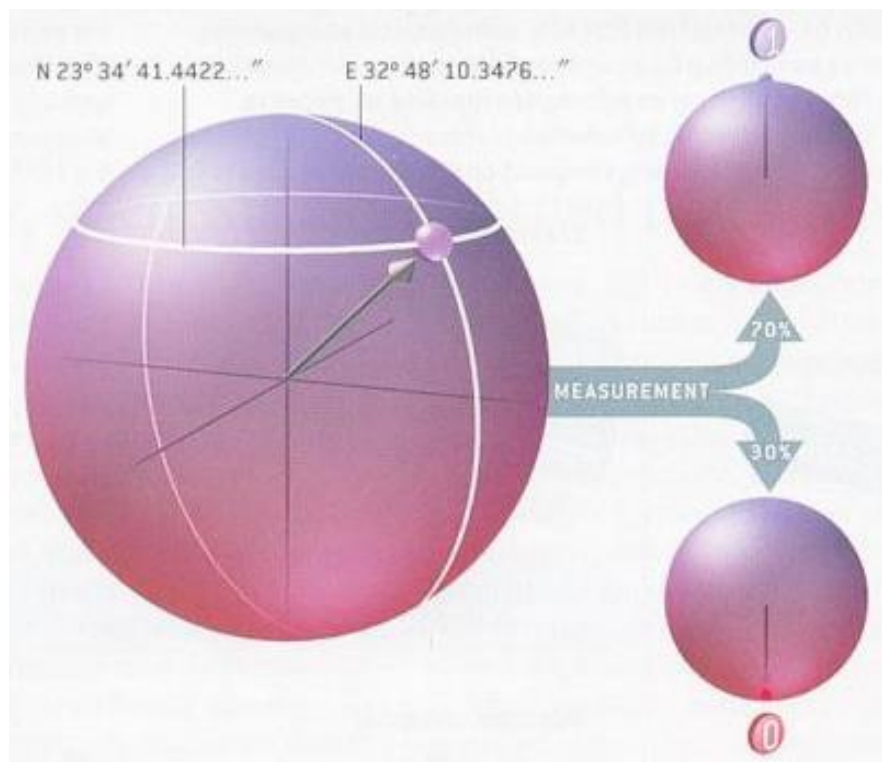


Figure 35: Quantum’s qubit continuum state vs classical bit information representation

3.3.8 Performance Evaluation

(i) Performance Evaluation Metrics for Base and WVE Models

Measuring the discrimination ability of a model is one of the important aspects of assessing its performance (Pearce & Ferrier, 2000). The metrics we used to evaluate the performance of our models are accuracy, precision, recall, f1-measure, Cohen kappa, receiver operating characteristics (ROC) analysis, with its associated area under the curve (AUC) (Brownlee,

2020b, 2020b; Hanley, 1989; Obuchowski & Bullen, 2018; Soleymani *et al.*, 2020). The basic performance metric Accuracy, to address the multi-classification problem precision, recall, and f-measure is the most widely used metrics for evaluating the performance of machine learning classifiers. Although past performance may not be indicative of future results, the mentioned metrics formed the common ground for determining how well the developed classifiers might perform in the future. These metrics were computed by using the powerful Scikit-learn (Sklearn) library for ML applications in python (Pedregosa *et al.*, 2011), which consists of a wide range of ML classification metrics.

Contrary to supervised regression learning which utilizes the Root Mean Square Error (RMSE), as a measure of model performance, classification learning uses accuracy, precision, recall, f-measure, kappa statistic, and Area under the ROC curve (AUC) which are derived from the basic confusion matrix metrics shown in Table 8, Where, True +ve, False +ve, True –ve, and False –ve are the respective numbers of true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Table 8: The four basic confusion matrix metrics

Classes	Test result +ve	Test result –ve
Actual +ve	True +ve(TP)	False -ve (FN)
Actual –ve	False +ve (FP)	True -ve (TN)

One of the keys behind modeling with ML algorithms is to have a relevant featured dataset to use in running experiments that aim to achieve improved performances by optimally tuning the training parameters as calculated from the confusion matrix (Hall *et al.*, 2015; Osisanwo *et al.*, 2017; Witten *et al.*, 2016). Table 9 summarizes all the metrics used to evaluate the individual ML base models and the resultant WVE performances.

Table 9: Some supervised learning performance metrics

Metric	Description	Calculation
Classification Accuracy(CA)	Percentage of correct predictions where the top class (the one having the highest probability), as indicated by the model, is the same as the target label as annotated beforehand by the authors. For multi-class classification problems, CA is averaged among all the classes. CA is mentioned as a Rank-1 identification rate (Hall <i>et al.</i> , 2015).	$CA = \frac{(TP + TN)}{(TP + TN + FP + FN)}$ Equation (16)
Precision (P)	Fraction of TP from the total amount of relevant results, i.e. the sum of TP and FP. For multi-class classification problems, P is averaged among the classes.	$P = TP / (TP + FP)$ Equation (17)
Recall (R)	Fraction of TP from the total amount of TP and FN. For multi-class classification problems, R gets averaged among all the classes.	$R = TP / (TP + FN)$ Equation (18)
F1 Score or F-Measure (F1)	Harmonic means of precision and recall. For multi-class classification problems, F1 gets averaged among all the classes. It is mentioned as F-measure (Minh <i>et al.</i> , 2017).	$F1 = 2 * (TP * FP) / (TP + FP)$ Equation (19)

(ii) Performance of the proposed 1EXP (-) Z⁺IT-ASMPSS-BEO-ISWVE

To evaluate the performance of the proposed 1EXP (-) Z⁺IT-ASMPSS-BEO-ISWVE, an asymptotic analysis of its 1EXP (-) Z⁺ initial term-based search space sequence formulation function was performed to determine its computational complexity as a means to understand the algorithm efficiency. Also, the hardware clock cycles based on execution times, and size complexity were obtained during the proposed algorithm execution by profiling its search space function, to evaluate its performance.

The proposed algorithm's effectiveness in producing effectual search spaces was evaluated by computing accuracies. Whereas, area under the receiver operating characteristic (AUC-ROC) curve analysis implementation result was used as the main measure of effectiveness by observing the scores of the various WVE that are estimated from the formulated previously deemed effectual search spaces that resulted from the proposed 1Exp (-) Z⁺ initial-term based arithmetic sequences search spaces with accuracy maximization as an objective function. Whereas the accuracy of a classifier is the most spontaneous measure for performance, we used

it herein as baseline performance only, for both the base models and the resultant 2S-HHEC, as an intuitive measure of model performances and optimization criteria, and it was not used as the key leading model performance measure. Instead, the area under the receiver operating characteristic curve (AUC–ROC curve) shown in equation (20), for a multiclass problem was used to select the final VWE models based on the quality of their predictions.

$$AUC = \frac{1}{c(c-1)} \sum_{j=1}^c \sum_{k>j}^c (AUC(j|k) + AUC(k|j)) \quad (20)$$

As asserted by Pintelas and Livieris (2020), “*c denotes the total number of classes, AUC (j / k) represent AUC having positive class j and negative class k*” (Yang *et al.*, 2019), was adapted using the one vs rest (OVR) and one vs one (OVO) with ‘multiclass’ arguments, where the one-vs.-one, and one-vs.-rest average ROC–AUC scores for class labels were calculated. The final ROC analysis results were plotted as ROC-AUC curves by using the false positive rate (1-specificity) against the true positive rate (sensitivity) (Carter *et al.*, 2016; McClish, 1989; Obuchowski & Bullen, 2018; Okey *et al.*, 2022; Yang *et al.*, 2019).

Therefore, for detailed investigation of the predictive quality of the resultant 2S-HHEC, the receiver operating characteristics (ROC) analysis was used whereby the measure of its ability to increasingly provide correct predictions could be determined through a 2 dimensional (x,y) plane ROC curves visualizations of its true positive rates (sensitivity) vs false positive rates (1 – specificity), in such this helped us determine how correctly each of the ‘low’, ‘medium’, and ‘high’ soil fertility classes may correctly be predicted in future unseen data or new observation. While ROC curves are useful for quick visualization of the classifier’s quality, the area under the curve (AUC) of the ROC (AUC-ROC) was obtained to determine the actual value of the area under the ROC curve. Achievements of accuracy close to 100% are desirable, although in imbalanced multi-classification tasks F1-measure under precision and recall are the most suitable measures of classifiers' discrimination ability. Cohen kappa was used to measure the inter-rater agreement, Kappa’s value closing to one shows great agreement between data collectors, to zero shows no agreement, it was used here to determine how reliable is it that the model was trained with the right data geometry, i.e. the level of agreement between the variable used to train the model and the data it was supplied with.

3.3.9 Validation and Utility of the Model

The metric used to validate the resultant WVE model performances was the percentage correct classifications, while maize grain yields were used to measure the utility of the model in terms of agricultural productivity as highlighted in subheading i) and ii) sections.

(i) Percentage Correct Classifications

The percentage correct classifications were used to validate the developed WVE model by using the soil laboratory-based test results.

(ii) Maize Plantation Fields Experimentations Grain Yields

DSR requires that the utility of the research contribution should be validated, for instance with the corresponding problem environment(s) through various methods such as field experimentation. For that purpose, we performed field experimentations to validate the utility of the proposed 2S-HHEC model as DSR artifacts through maize plantation experimentation with the incorporation of the model-based experimentation filed soil fertility status prediction information to examine its utility. Maize plantation field experimentation to obtain maize harvests amounts based on the model, and basic recommendations, as well as Adhoc soil fertility management practice. Table 5 presents the plantation experiment planning details on an acre of farmland, whereby that was divided into three subsections labeled “M” to denote the model-based predictions information farmland fertility management plantation section, the next labeled as “B” for the basic and blanket recommendations applied uniformly across the plantation section. The last is “A” for ad-hoc controlled plantation, where no measurements, treatments, or any major management practice was applied other than cultivating that land, seedling, weeding, and harvesting, while for the model-based we measured the chemical nutrients values and treating those areas where fertility was predicted low or medium before plantation, and blanket recommendations were applied for the other remaining respective section. Each of these sub-plantation sections was contained in blocks stratified across each of the four quarters of the acre. Each of these sections, i.e. model-based predictions based controls (M), basic and blanket recommendations (B), and ad-hoc control (A), occupied 33.33% of the 1-acre plantation area. Each quarter was cut into 16 line sections for each management label making a total of 48 lines in each quarter, and 196 lines for the entire farm (see Table 10). As such each control there could be obtained from 64 samples, these of which were measured for only those sections to be considered for model-based predictions of fertilization and

management, and the other 128 samples were left as is with 64 of them managed with the basic blanket recommendations, and the other 64 samples section were untreated. To assume a constant availability of water supply to eradicate its variability effects in our study results, we had in place a furrow irrigation facility.

Table 10: Experimental plantation plan

Lines sections	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
16	M	B	B	M
16	A	M	P	B
16	B	A	M	P
Total	48	48	48	48

Finally, as a measure of the proposed models' predictive information utility on maize field plantation experimentation, we calculated the profiles of the number of maize harvests in tons per acre following the maize for the model-based predictions based controls (M), basic and blanket recommendations (B), and ad-hoc control (A).

CHAPTER FOUR

RESULTS AND DISCUSSION

This section presents the results obtained from the experimentation of the developments in this research. These include results from preprocessing of the dataset used in the experiments, evaluating the implemented ML individual base models, the developed novel brute exhaustive search procedure optimization algorithm, the resultant WVE, and its validation as well as evaluation of the model's utility by using its sample fertility status predictions information on a maize field's experimentations.

4.1 Data Pre-processing Results

4.1.1 Descriptive Statistics of Used Dataset

The section presents the data results obtained from the proposed research development of a machine learning ensemble model for high-performance soil fertility status prediction. Table 11 shows the statistical description of the data used in the experiment showing count, mean, standard deviation (std), min, 25%, 50%, and 75% percentiles, and max for each attribute in the dataset.

Table 11: Statistical description of the used agricultural soils and yield data

	OC	pH	EC	TN	P	Ca	K	Mg	Na	S	Mn	Al	Zn	Fe	B	M_Yld
Count	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260	6260
Mean	1.875	5.912	1.412	0.080	5.003	1.487	0.147	0.180	4.184	56.738	8.391	10.131	0.814	45.689	0.013	1.658
Std	0.804	0.516	2.493	0.043	2.854	1.732	2.450	0.484	22.17	27.196	5.325	29.678	2.314	32.956	0.019	0.679
Min	0.256	4.257	0.001	0.008	0.062	0.005	0.002	0.001	0	0.365	0.006	0.01	0.009	0.401	0.002	0.63
25%	1.211	5.547	1.071	0.044	3.135	0.388	0.02	0.050	0.081	37.616	5.427	0.439	0.476	35.382	0.008	1.22
50%	1.667	5.904	1.368	0.064	4.571	0.960	0.031	0.105	0.207	53.687	7.335	0.566	0.584	41.988	0.011	1.62
75%	2.543	6.274	1.716	0.116	6.27	1.985	0.047	0.200	0.610	73.754	10.312	0.821	0.744	48.895	0.014	2.11
Max	4.211	7.457	139.46	0.206	37.721	27.021	82.313	17.842	803.93	310.55	103.99	554.2	51.441	1068.5	1.003	4.39

As can be observed from Fig. 36 of the comparison of observations to features ratio of the used dataset in similar ML implementations, 6260 observations which were collected for the 15 features, represented a good observation to features ratios for training and testing the ML model(s), as compared to other studies.

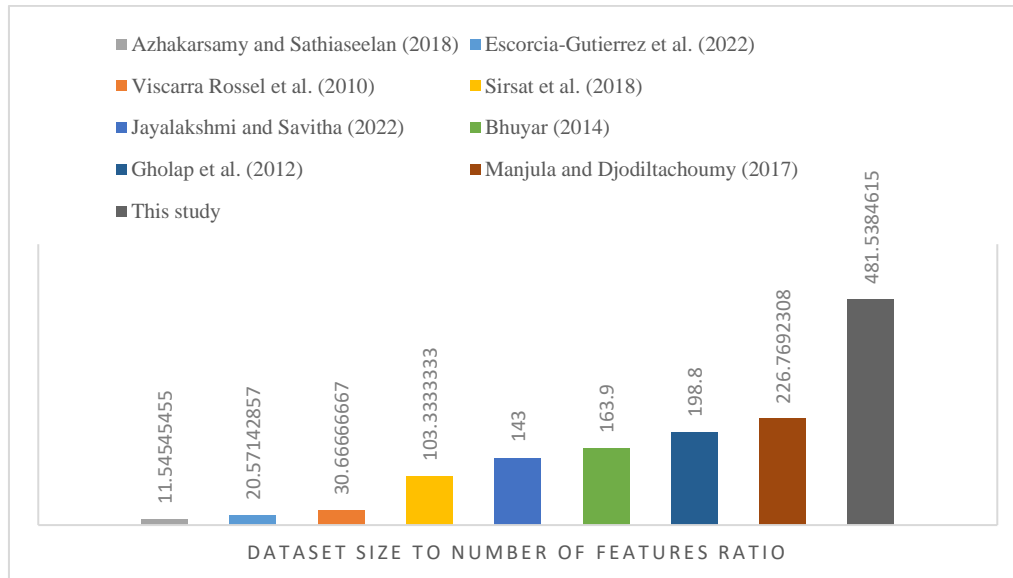


Figure 36: Comparison of observations to features ratio of the used dataset in similar ML implementations

4.1.2 Quality of Dataset

The QQ plots for the soil chemical properties data were plotted to visualize and determine if they are either randomly or normally distributed to perform informed data treatment accordingly. It could be found that the pH variable followed a normal distribution with points lying across the $y=x$ axis of the QQ plots (Chan, 2022; Varshney, 2020). Sulphur and phosphorus were right-skewed, while Organic carbon exhibited high kurtosis as it crossed the $y=x$ axis in the middle (see Figs 37 to 40). Therefore, outliers and extreme values were identified by using the IQR method and removed before some of the data's normalizable random variables were transformed before the application of any ML algorithm for modeling our stated problem scenario solution(s).

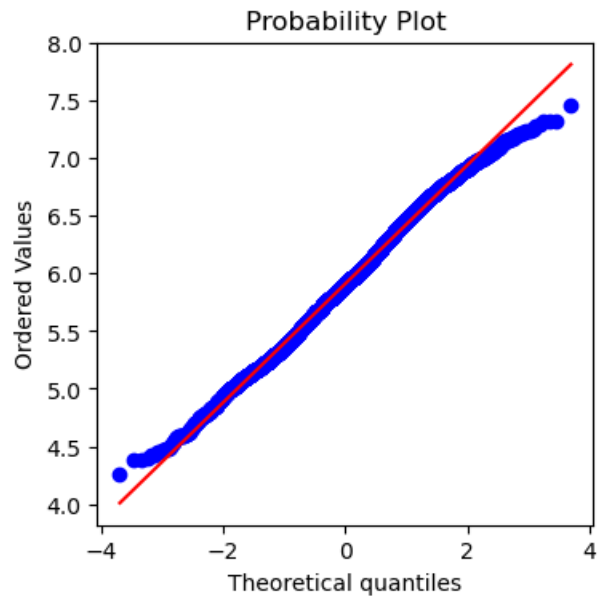


Figure 37: pH QQ Plot

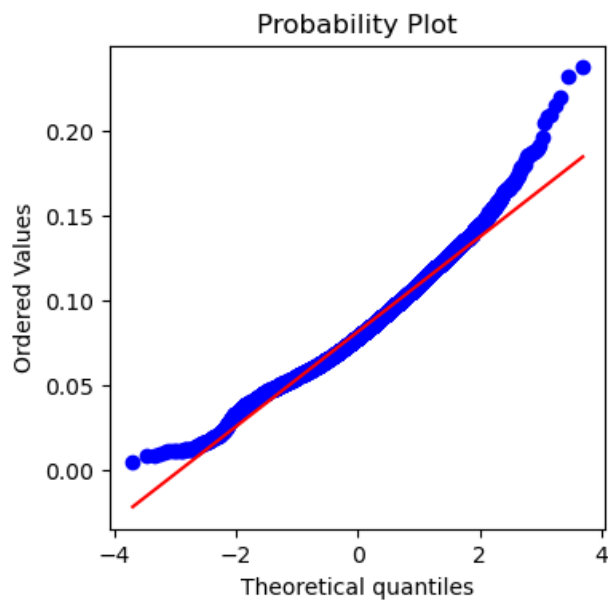


Figure 38: S QQ plot

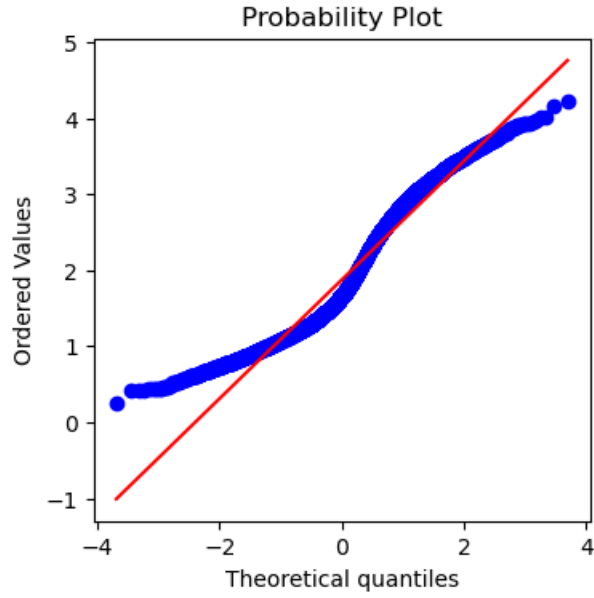


Figure 39: OC QQ plot

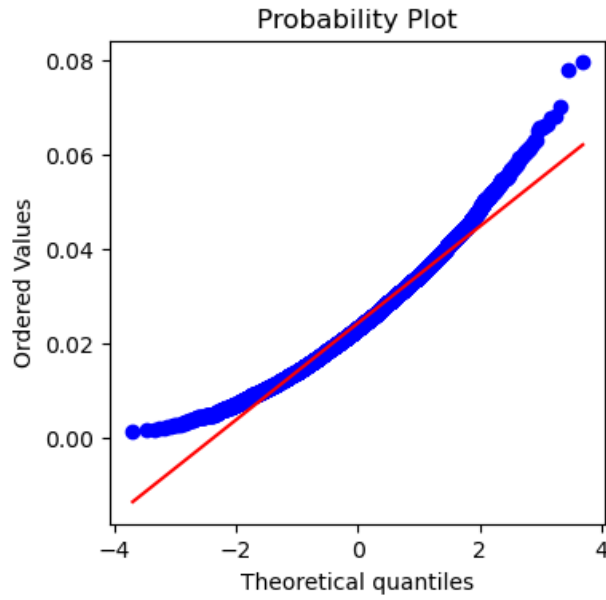


Figure 40: P QQ plot

4.2 Modeling and Optimization Results

4.2.1 Soil Fertility Index derivation

Results of modeling the soil fertility characteristic index in the data-preprocessing phase are shown in Figs. 41 to 49. Three optimum groups were obtained, which resulted in three distinct soil fertility status indices. While these results conform to those in studies that used a similar

number of class targets, that is three, they differ from the majority of other studies that made use of only two (2), four, or five soil fertility status target classes.

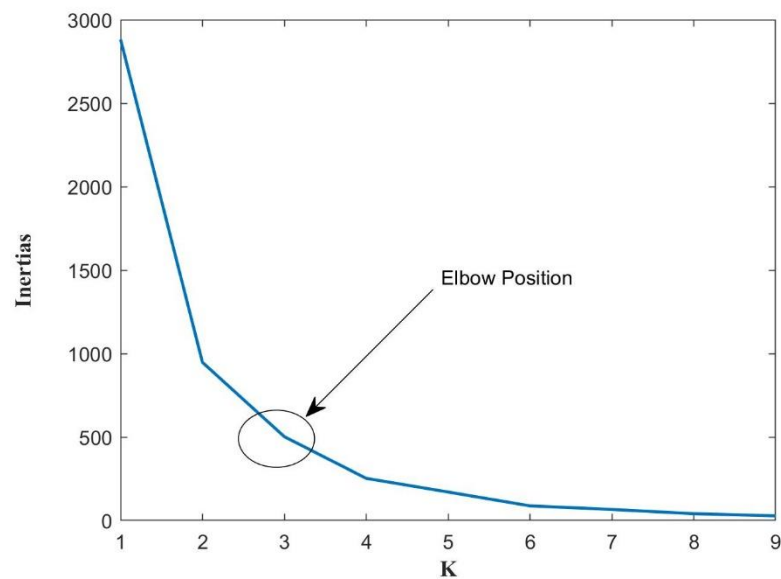


Figure 41: Knee elbow = 3

Figure 42 shows test results for the analysis of variance (ANOVA) between the different formed fertility groups in the data with the null hypothesis “the clusters groups are not similar” tested true meaning the clusters are different.

ANOVA (Tukey Test) results on the data groups:---

	group1	group2	meandiff	p-adj	lower	upper	reject
0	high	low	0.5791	0.0000	0.4468	0.7114	true
1	high	medium	0.4494	0.0000	0.3139	0.5849	true
2	low	medium	-0.1297	0.0000	-0.1806	-0.0789	true

Figure 42: Test Results for the Analysis of Variance between different fertility groups

Figure 43 is the plot of the ANOVA results, which were found using the Tukey HSD Test.

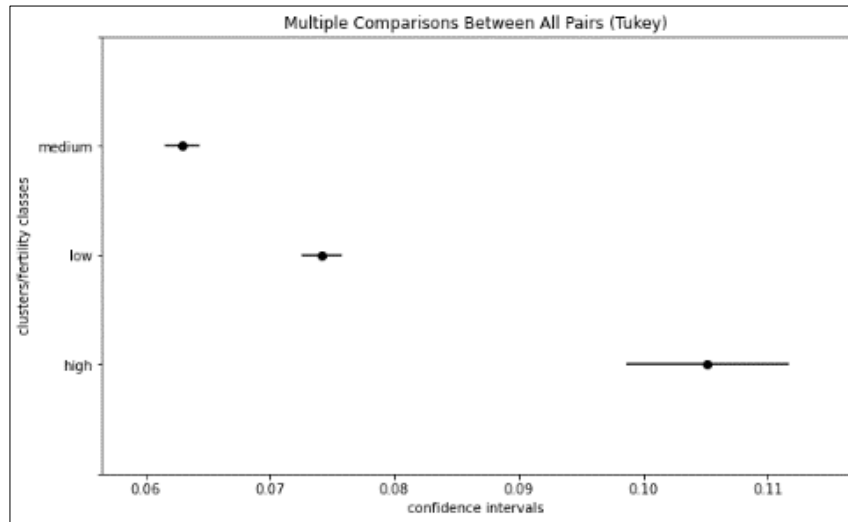


Figure 43: The ANOVA (Tukey Test) results

The groups were visualized in plots as shown in Figs. 44, 45, 46, and 47 of the pH and Organic carbon, vs fertility index dimension. Followed by respective visualizations of organic carbon, calcium vs fertility index clusters visualization, Organic carbon, Phosphorus vs Fertility index clusters visualization, and finally but not least, of organic carbon, electrical conductivity vs fertility index clusters are depicted.

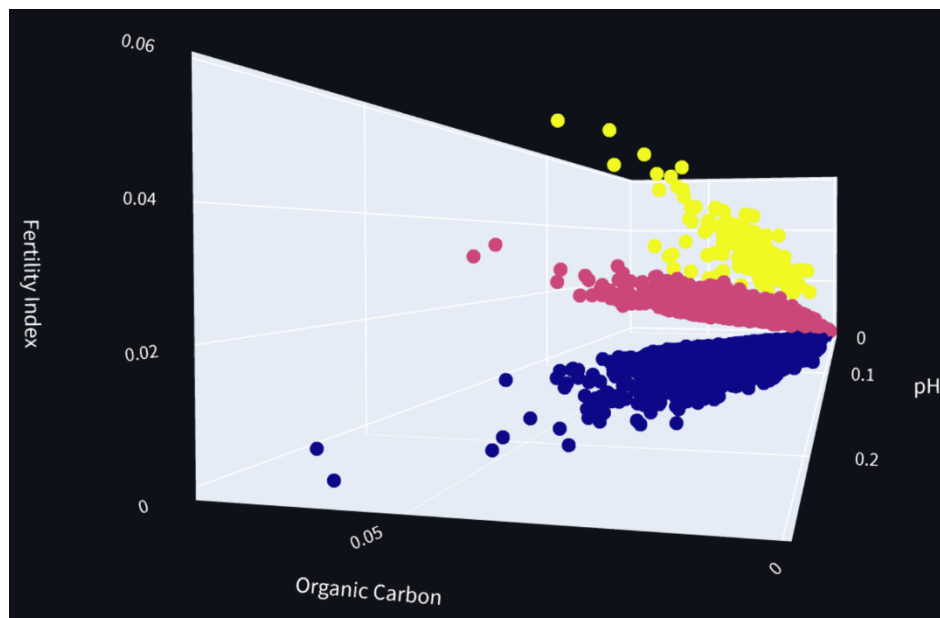


Figure 44: Organic carbon, pH vs Fertility index clusters visualization

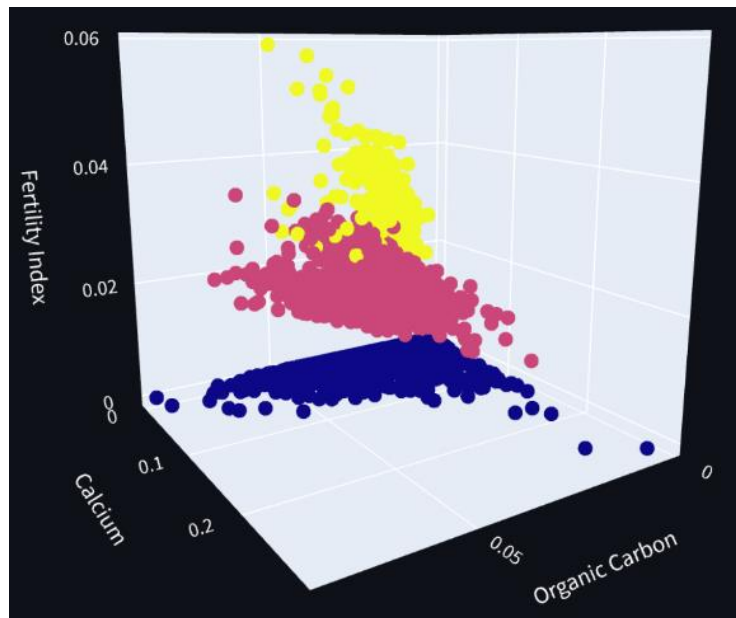


Figure 45: Organic carbon, Calcium vs Fertility index Clusters visualization

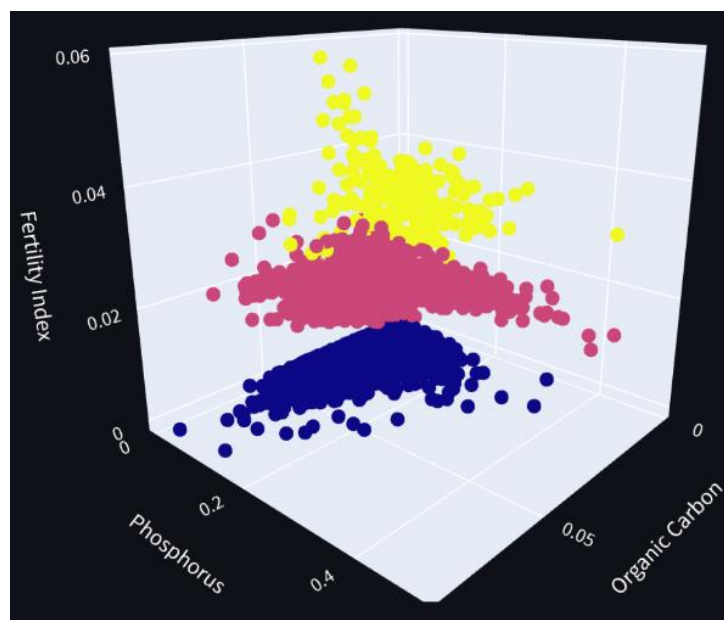


Figure 46: Organic carbon, Phosphorus vs Fertility index Clusters visualization

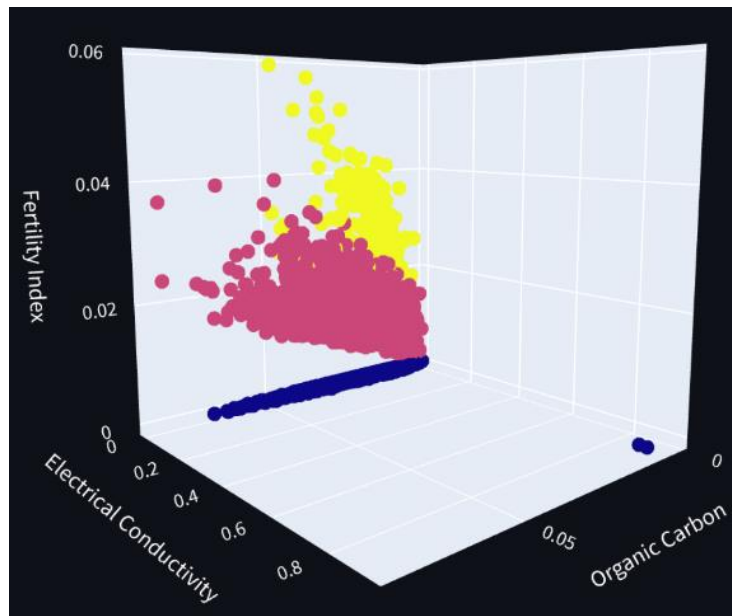


Figure 47: Organic carbon, electrical conductivity vs fertility index clusters visualization

Figure 48 shows the distribution of the fertility classes. It could be seen that majorly parts are lowly characterized by soil fertility.

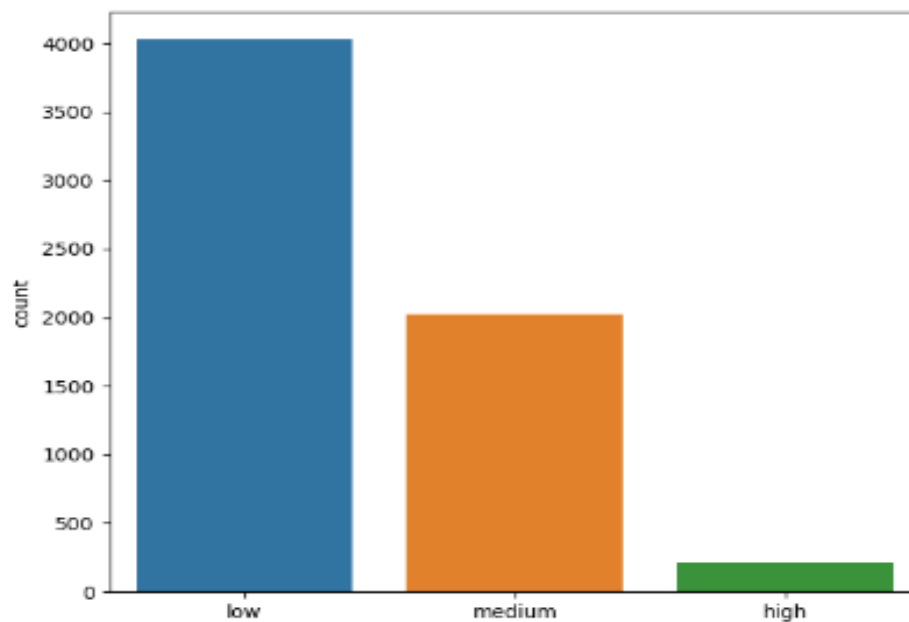


Figure 48: Soil Fertility classes (label) distribution

Figure 49 displays the correlation heatmap showing the percentage correlation of the features in the data as well as the formed target classes low represented as 0, medium as 1, and high as 2. The final dataset for evaluating different ML algorithms classifiers was obtained by removing Magnesium as it highly correlated with potassium at 70% correlation and Total nitrogen had an 89% correlation with organic carbon.

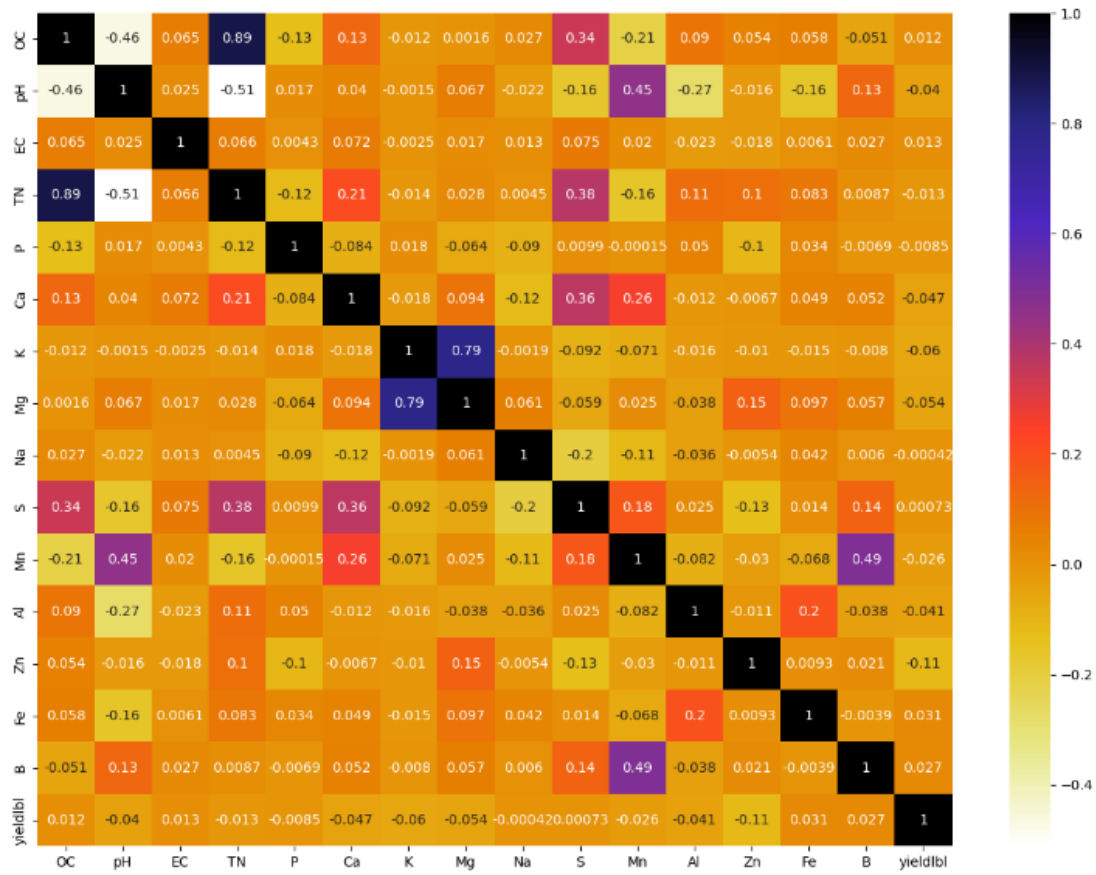


Figure 49: Correlation heatmap of the soil chemical properties

4.2.2 Base Models: Performance Evaluation

After developing the various heterogeneous hybrid base classifiers models (HHCM) for predicting soil fertility statuses, whereby the performances of K-Nearest Neighbor hybrid (KNN-H), Support Vector Machine Hybrid (SVM-H), Decision Tree Hybrid (DT-H), Random Forest Hybrid (RF-H), Adaptive Boosting Hybrid (AdaB-H), Naïve Bayes Hybrid (NB-H), and Gradient Boosting Hybrid (GB-H) were evaluated by using test data to determine Table 12 displays the results obtained from the evaluation.

Table 12: The HHCM performance

HHCM	Precision					Recall				F1Score			
	Accuracy	Low	Mid	High	Weighted	Low	Mid	High	Weighted	Low	Mid	High	F1_Score
KNN-H	0.90*	0.82	0.97	0.62	0.91	0.93	0.88	1	0.9	0.87	0.92	0.76	0.90**
SVM-H	0.91*	0.85	0.93	0.86	0.91	0.86	0.92	0.98	0.91	0.86	0.93	0.91	0.91**
DT-H	0.87*	0.79	0.94	0.56	0.88	0.87	0.87	0.9	0.87	0.83	0.9	0.69	0.87***
RF-H	0.91*	0.89	0.94	0.66	0.92	0.87	0.93	0.97	0.91	0.88	0.94	0.78	0.92**
AdaB-H	0.52*	0.43	0.75	0.1	0.63	0.48	0.53	0.6	0.52	0.45	0.62	0.17	0.55
NB-H	0.11	0.3	0.81	0.04	0.62	0.13	0.06	0.9	0.11	0.18	0.11	0.07	0.13
GB-H	0.93*	0.89	0.95	0.82	0.93	0.89	0.94	0.97	0.93	0.89	0.95	0.89	0.93*

From the results, it could be seen that most of the individual classification hybrid models demonstrated good prediction performance on test data, with gradient boosting hybrid classifier scoring the highest accuracy of 93%, as compared to RF-H, SVM-H, and KNN-H models that achieved respective predictive accuracies of 92%, 91%, and 90% for (see Table 10). Comparing these results with similar objective model accuracy results of authors Chaudhari *et al.* (2020) who implemented a Decision Tree classifier for predicting soil fertility with an accuracy of 60%, our results were 30% better.

4.2.3 Computational Complexity: Efficiencies

An Asymptotic Analysis was performed for both the 1EXP (-) Z^+ initialization function and its brute exhaustive search-based optimization algorithm.

(i) The 1EXP (-) Z^+ Initialization Function Asymptotic Analysis

An asymptotic analysis of the proposed algorithm's overall sequences generation function expression in equation (10) could further be evaluated to determine the mathematical validity of the 1EXP (-) Z^+ based WVE computation in equation (14). As shown in Fig. 50 of the derived 1EXP (-) Z^+ initial term-based arithmetic sequences formulation function expressions 3-D graphical display of its valid computational space, portrayed asymptotic optimality to the WVE constrained boundaries in equation (2).

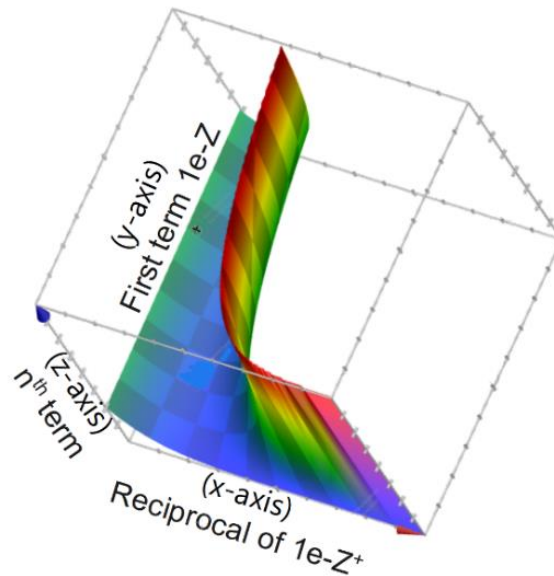


Figure 50: The 1EXP (-) Z^+ based Sequence Initial term function asymptotic optimality to WVE weights constraints

Whereas it can be observed the sequences initial term values represented by the y-axis are asymptotic to zero in such the $1\text{EXP}(-)Z^+$ based initialized values may get smaller as much as but never equal to 0, hence the weights greater than 0 constraints is always maintained through that presented asymptotic characteristic, in turn, the size of the sequence may grow larger to as much as the reciprocal of the $1\text{EXP}(-)Z^+$ as read from the x-axis based on the initialized sequence's first term on the y-axis. Based on those facts, the proposed function is considered mathematical valid for an optimal algorithmic system computational implementation. At that juncture a function for explicitly formulating values as weight coefficients $W[k][n]$ which satisfy WVE weights constraints in equation (2) could be derived based on the proposed $1\text{EXP}(-)Z^+$ initial term arithmetic sequences, then a brute-exhaustive optimization could be applied to search one optimal combination set, hence that function provides for an algorithmic computational implementation. The proposed sequences formulation function is asymptotic optimal to the WVE constraints in equation (2) and as it is integrated into the proposed weights coefficients values formulation algorithm's matrix function with expressions in equation (10), it can then be deduced that this function can be mathematically valid for computational implementation to generate WVE combinations weighting values for use as the result of a system of linear equations represented as $Y[k][d]$ as combinations predictions.

(ii) The $1\text{EXP}(-)Z^+$ -ASMPSS Complete Algorithm

The $1\text{EXP}(-)Z^+$ -ASMPSSA_BES_ISWVEO computational complexity was then asymptotically analyzed by calculating the proposed algorithms instructions lines asymptotic execution time as follows: delineating the complexity times from the above-proposed algorithm pseudo code to calculate the asymptotic total complexity time, we obtain the total complexity to be as *Total Complexity (TC)* = $F(Z) = \{1\} + \{1\} + \{1\} + \{1e^{-Z^+}\} + \{1\} + \{1\} + \{((1e^{Z^+}) + ((1e^{-Z^+}) * (N-1)))\} + \{1\} + \{1\} + \{1e^Z\} + \{1\} + \{1\} + \{1\} + \{1\} + \{1\} + \{1\}$

$$TC = \{11\} + \{1e^{-N}\} + \{((e^{-N}) + ((e^{-N}) * [(N - 1)]))\} + \{1e^N\} \quad (15)$$

Deducing from the highest order term of the algorithm's derived total complexity TC in equation (15), it can be seen that $\{1e^N\}$ is the highest-order term worst-case scenario (Big O), this will represent the upper bound of the algorithm running time. Therefore the algorithm has

a *worst-case scenario* exponential complexity of $O(1e^N)$. When this type of computational time complexity might be undesired in cases where search space precision grows so large, the upper bound running time could even fast be reached when the search spaces are integrated into the brute exhaustive-based search heuristics algorithm execution that would mainly arise from the size of ensemble base expert predictions to be weight estimated during optimization. Whereby, algorithm execution acceleration procedures namely, the constraining of search spaces with weights points, coupled with the vectorization of data structures thereof, and computation on reasonable computational hardware resources were used to facilitate for rapid execution of the algorithm computations in attempts to provide the algorithm execution run time minimization.

Thus, in this study, it could be concluded that the proposed 1EXP (-) Z^+_{IT} -ASMPSS-BEO-_{IS}WVE algorithm exhibited an exponential time complexity running time. While, this type of computational time complexity might be undesired in cases when the search space precision processing requirements grow large, whereby the search spaces constraining procedure, coupled with the weight search spaces data structures vectorization, and use of core i8 with 64GB RAM computation hardware facilitated for accelerated processing of the algorithm computations as key means to minimize its running time.

4.2.4 EXP (-) Z^+_{IT} -ASMPSS-BEO-_{IS}WVE Optimization Results

Following executions of the proposed 1EXP (-) Z^+_{IT} -ASMPSS-BEO-_{IS}WVE algorithm, its efficiency and effectiveness performances were observed, logged, and presented here in this research. Results of the algorithm efficiency are presented in Table 13 which shows hardware cycle-based execution times in formulating relevant search spaces for references $Z^+ = 1, 2$, and 3 corresponding to 1EXP (-) Z^+ based respective arithmetic sequences precision's factors Precision 0.1 with 10 possible points or sequence length, 0.01 having 100, and 0.001 producing 1000 sequence terms to be considered as weights. Execution times were increasing not only in: a) the formulation of the sequences from 0.000120 seconds in search space 1 to 1262.4213 seconds in 3, as well in constraining the valid search spaces $C_Feasible_S$, and in the overall total optimization execution time Opt_T of 90.3207 seconds to perform the search in search space 1, whereas 2701.32 seconds were taken to search space 2 with deeper precision 3 for factor 0.01 before to reach hardware limitation after 4834.00 seconds in search space reference 3. But, b) also in with an increase in the search space precision referenced by Z^+ , but most critically with an increase in the consequent procedures of $C_Feasible_S$, and time Opt_T of

90.3207 seconds in search space 1, 2701.32 seconds in 2. However, whereas in search spaces with precision 0.1 and 0.01, the proposed algorithm showed stable execution to convergence in the generation of potential systematic solutions for searching optimality, on the contrary, in search space reference $Z^+ = 3$ with precision 0.001 the execution was halted after approximately 8450 seconds.

Table 13: Search space precisions, formulation, and containment and optimization times

Space ref (Z^+)	Precision	Points	Formulation	Time in seconds(S)	
				C_Feasible_S	Opt._T
1	0.1	10	0.000120	0.0536000	90.3207
2	0.01	100	1.750000	170.07232	2701.32 (45 min)
3	0.001	1000	321.0000	1262.4213	8234*H/W Lim

Concerning the hardware execution times, as could be seen from the execution times results in Figures 51 and 52 of the respective Hardware clock time in search space precision 1 and 2, and Hardware clock time in search space 3 execution before reaching processor limitation. It could be observed that the algorithm took 2700.32 seconds (45 minutes) total optimization time optimize from the combinations in search referenced by $Z^+ = 2$ with precision scale 0.01 thus 100 different weighting values, unlike in space reference $Z^+ = 1$ with precision factor 0.1 having fewer points, or possible weight values, specifically 10 points or terms, were it took approximately 90.3207 seconds (1 minutes and 30 seconds) only, this of which could be explained by the increase in number of weight points to 100 in $Z^+ = 2$ hence explaining the significantly large difference in between these times. Also, in $Z^+ = 3$, actual misbehavior started to exhibit with a premature optimization execution halted before completion of formulation of search spaces, only to be able to save a few which were later on executed in isolation mode as they could not complete processing under normal hardware capacity, whereby 8234 seconds, which is approximately 2 hours and around 17 minutes were taken, independent of the briefed 1EXP (-) Z^+_{IT} -ASMPSS-BEO-IsWVE package to return an optimal WVE configuration set with 5 base models across that much deeper 0.001 precision search space.

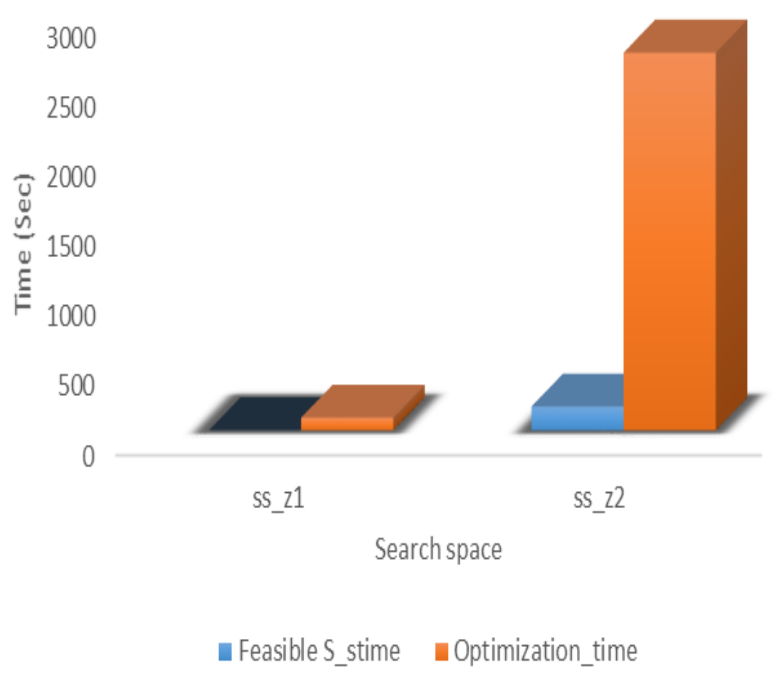


Figure 51: Hardware clock time in search space precision 1 and 2

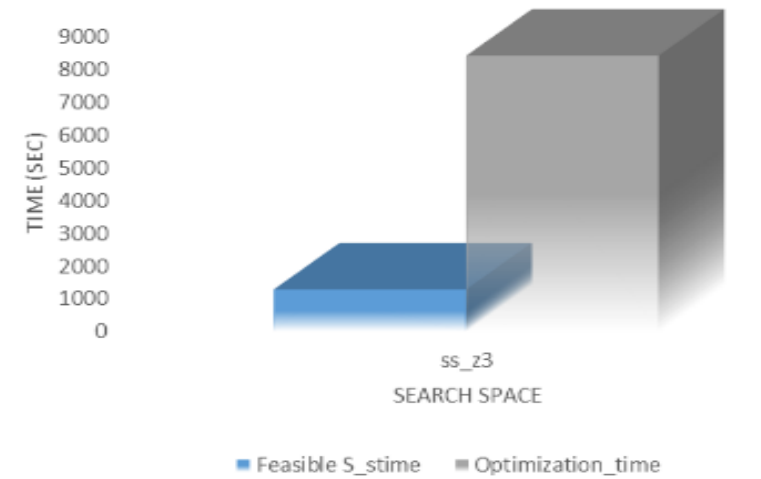


Figure 52: Hardware clock time limitation in search space 3

Figure 53 displays the results of the proposed 1EXP (-) Z^+_{IT} -ASMPSS-BEO- IS_{WVE} sequences formulations and optimization Algorithm Efficiency in the most stable search space reference in terms of total hardware execution time profile in stable search space reference $Z^+ = 2$, and memory consumption by the algorithm which could be observed to be approximately 85 MiBs,

this which was expended in less than 2 seconds to formulate the sequences in for search space with reference $Z^+=2$ having precision factor 0.01, in Core i8 64 GB RAM, which maybe reasonable in WVE optimization procedure.

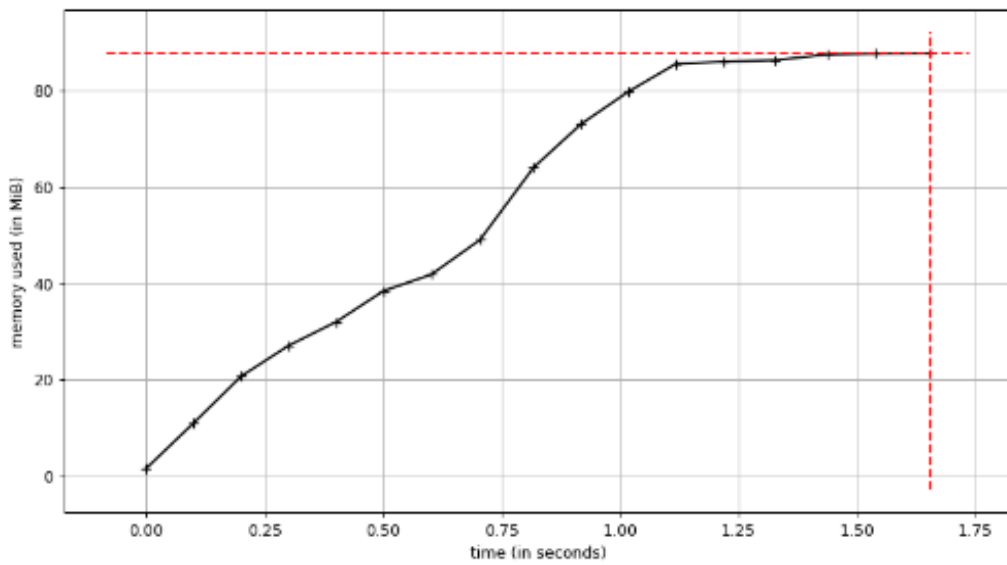


Figure 53: 1EXP (-) Z+IT-ASMPSS-BEO-ISWVE sequences formulations and optimization Algorithm Efficiency in the most stable search space reference $Z^+ = 2$

It could be observed that the total optimization time taken to search for solutions varied across searches spaces and these increased as the search space's precision increased from search space 1 through 3 as shown in Fig. 54. This is explained by the increase in iterations taken to execute the implemented search procedures due to increase in the number of search points in search space 1, 2, and 3.

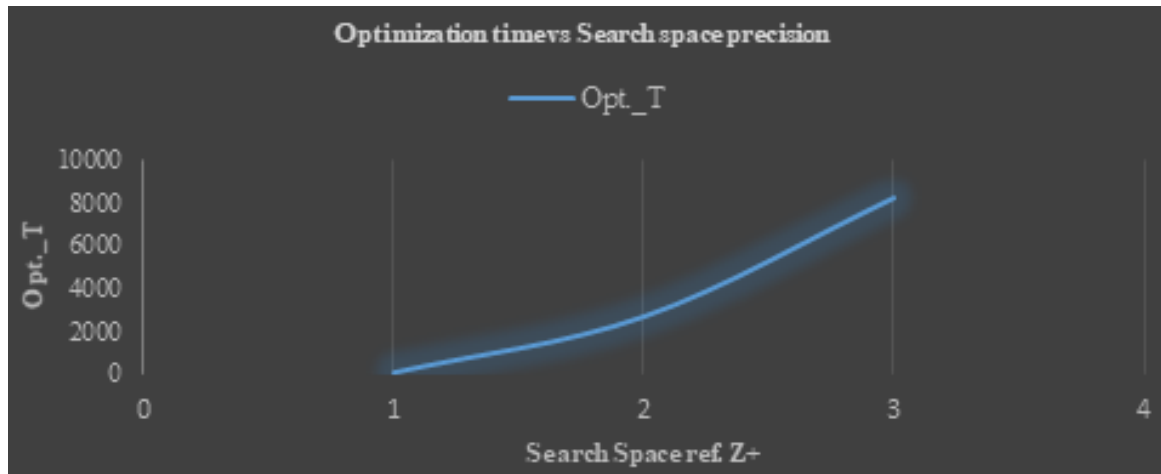


Figure 54: Total optimization time in search spaces

4.2.5 The 1EXP (-) Z^+_{IT} -ASMPSS-BEO- $ISWVE$ Effectiveness

The proposed 1EXP (-) Z^+_{IT} -ASMPSS-BEO- $ISWVE$ was highly effective in formulating multi-precision 1EXP(-) Z^+ based sequences that were processed to generate search spaces with varying combinations sizes in both search spaces 1 and 2, of which executions across referenced to these spaces were observably converging following the execution of the proposed implementation a countless number of times. With 10 different sequence values in search space one (1), 100 in two (2), and 1000 in search space three (3) where the experimental core is 64 GB hardware capacity limitation was reached to invoke the termination criteria, as a result forming an incomplete search space which was stored in log files. Among other reasons, that could be explained by IEEE 754 standard for FPN system's FPA requirements specifying hardware's math co-processor word bit size memory limitations for FPA (Committee, 2019).

As annotated by the search space domain 2 filtered combinations plot in Fig. 55. It can be seen that, unlike in search space 1, where five thousand and forty (5040) combinations were initially generated and filtered expressively by using the WVE weights boundary constraints in equation (2) as a reduction strategy that lead into only twenty (20) candidate solutions, whereas these may be tractable by trial and error heuristic procedure, it would be a tedious task to do the same in search space 2, where the total number of generated combinations grew exponentially to one hundred and thirty-three thousand nine hundred and ninety-two (133 192) further filtered combinations of candidate solutions subsets which is a reduction from the initial formed ninety-four million (94 000 000) combinations due the maximum weight coefficients value being constrained to max of 1. Such amount of combinations would instead be challenging to formulate without a computational algorithmic implementation, such as the one we proposed herein to effectively find optimal weights configuration sets based on prediction accuracy performance maximization as objective criteria through brute exhaustive searching by considering the available hardware capacity.

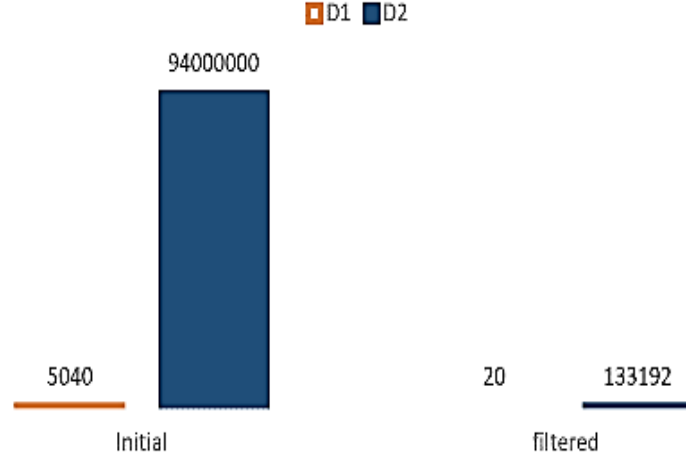


Figure 55: The WVE initial and filtered potential search combinations in the stable domain search space 1 and 2

To scrutinize the search space precision effect on the optimality based on accuracies of the various best WVE subsets, the proposed 1EXP (-) Z^+_{IT} -ASMPSS-BEO-ISWVE was executed in search spaces 1 and 2 with respective precision factors 0.1 and 0.01. And its partial logged combinations were processed independently of the package where it could not complete execution search space reference $Z^+ = 3$, with a precision factor of 0.001.

Table 14 shows the accuracies denoted as *Acc.*, macro, and micro ROC_AUC best scores that were achieved by the different WVE. After the various WVE subset combinations were generated and explorations by the proposed algorithm, based on search spaces precisions $Z^+ 1$ to 3, as well as the number of the different individual base models entailing to its heterogeneity. Results showed that the proposed algorithm had an outstanding performance in obtaining WVEs that had similar accuracies in the range of and even more as compared to results in some publications of similar objective models. As can be seen from Table 11, gradient boosting (GB), random forest (RF), support vector machine (SVM) and Knearest neighbor (KNN) WVE combination with respective weights 0.26, 0.01, 0.43, and 0.30 predict the status of soil fertility at an accuracy of 94% could be found in the stable optimization space 2, while the partial search space 3 combinations independent search execution could obtain a combination of gradient boosting (GB) with weight 0.187, random forest (RF) with 0.210, 0.175 for support vector machine (SVM), Knearest neighbor (KNN) at 0.321 with an additional heterogeneity by decision tree (DT) hypothesis weighted with 0.107, at an accuracy of 98.93 %. Whereas, those results were outstanding as the results in like in Jayalakshmi and Savitha (2022), unlikely the maximum accuracy observed in space 1 was 93% for both the combinations, at precision 2 with

scale 1, which could have accounted for the low accuracy therein unlike in the higher precision search spaces 1 and 2.

Table 14: Results of the effectiveness of the proposed search heuristic procedure

Spaces (Z^+)			1	2	3
Ens. ID	predictors	weights	Acc. (%)	Weights	Acc. (%)
1	<i>DT</i>	0.9	90.89	0.94	92.80
	<i>KNN</i>	0.1		0.06	
2	RF	0.9	90.89	0.81	92.80
	KNN	0.1		0.19	
3	RF	0.9	90.89	0.81	92.80
	SVM	0.1		0.19	
4	<i>SVM</i>	0.9	90.89	0.94	92.80
	<i>KNN</i>	0.1		0.06	
5	GB	0.9	90.89	0.84	92.80
	SVM	0.1		0.16	
6	GB	0.9	90.91	0.94	92.80
	RF	0.1		0.06	
7	RF	0.1	92.67	0.15	93.98
	SVM	0.6		0.57	
	KNN	0.3		0.28	
8	GB	0.1	92.67	0.21	93.98
	RF	0.6		0.55	
	KNN	0.3		0.24	
9	GB	0.1	92.67	0.31	93.98
	RF	0.6		0.27	
	SVM	0.3		0.42	

Un-generated subsets

Spaces (Z^+)		1		2		3	
Ens. ID	predictors	weights	Acc. (%)	Weights	Acc. (%)	weights	Acc. (%)
*10	GB	0.1	93.27	0.26	93.98		
	RF	0.2		0.01			
	SVM	0.3		0.43			
	KNN	0.4		0.30			
11	GB	0.5	93.17	0.36	95.02	0.187	98.93
	DT	0.1		0.12		0.107	
	RF	0.1		0.13		0.210	
	SVM	0.2		0.21		0.175	
	KNN	0.1		0.18		0.321	

As shown in Fig. 56 of the Search spaces precision effect on the WVEs combinations accuracies, It could be seen that the ensemble optimality was seen to be affected by not only its diversity, but also the size, which is a function of diversity in the sense addition of the later widens the ensemble heterogeneity which entails increased in its size, but most critically concerning the previous objective of our study it is affected by the search space as represented by weighting values precisions and scale.

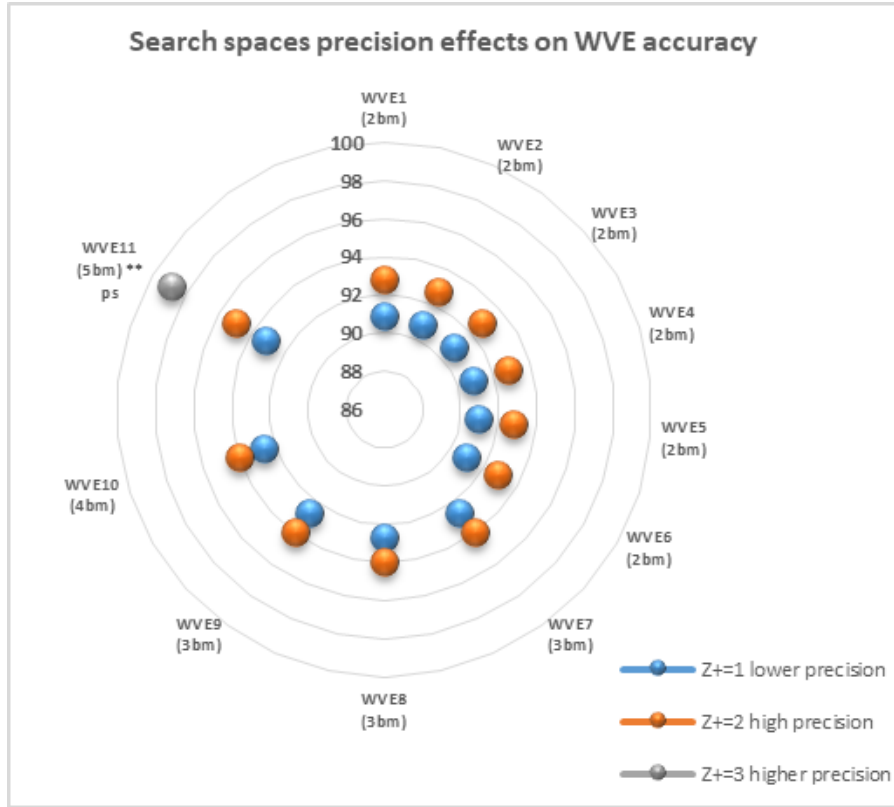


Figure 56: Search spaces precision effect on WVEs accuracies

This could be explained by the fact that better-refined solution values could be achieved in more refined also termed granular search spaces, were it was observed that lower search space precision led to lower WVE combinations accuracies, unlike higher precisions which have shown to weight WVE at higher prediction accuracies (see Fig. 61). This fact would represent a good indication on the effectiveness of the proposed $1EXP(-)Z^+$ IT-ASMPSS-BEO-ISWVE in generating search spaces as one of the key requirement for the successful execution of the consequent search procedure, as stated by Mouret and Clune (2015), that search spaces are a determinant factor as they have a significant effect in the overall optimization algorithm procedures implementation such as in finding WVE optimal subset, other than its diversity and constituting individual base model accuracies.

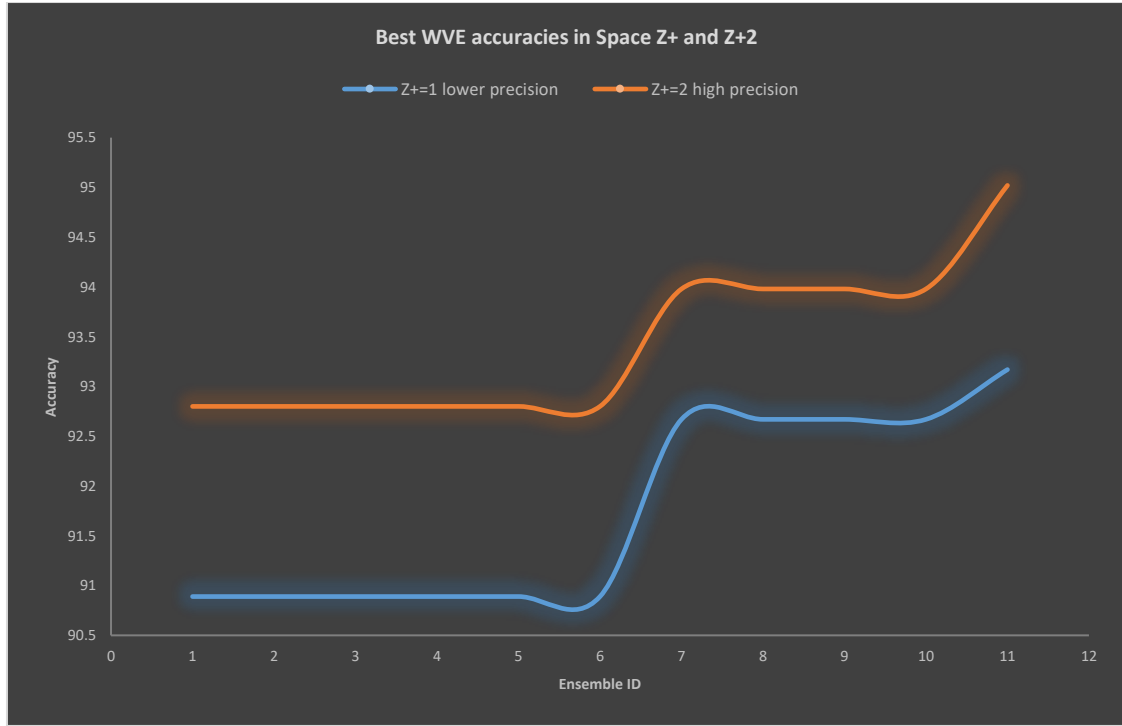


Figure 57: Best WVE accuracies in Space Z+ 1 and Z+ 2

Finally but not least, as an additional measure to enhance the optimal subset performance evaluation, the ROC curves were plotted with their corresponding AUC scores for some of the observed best accuracy combinations, as shown in Figs. 58, 59, 60, and 61 of the ROC plots and AUC results. Unlike the ROC plots and AUC scores results of algorithm execution in spaces 2 and 3, results based on using only two base classifiers combinations were highly uncondusive in space 1. As it could be observed, amongst the best, the two classifiers-based WVE combinations involving DT and KNN class probability predictions respectively weighted at 0.9 and 0.1, whose ROC plots and AUC scores are shown in Fig. 58 highly exhibited random guessing of the ‘*high*’ fertility class correct predictions with an AUC score of 58%, although these results are worse than those in other studies like in Rossel *et al.* (2010), outstanding optimization results could start to better be observed in space 2 with involvement of three base classifiers combinations as portrayed in Fig. 59 of the ROC plots and AUC scores of the three classifiers based RF, SV and KNN WVE combination with respective optimal weights 0.15, 0.57, and 0.28, whereby the combination could exhibit an increasingly ability in predicting all the fertility classes *high*, *medium*, and *low* correctly at respective AUC scores of 62%, 70%, and 71%, as well as average macro and micro respective 68%, and 86% AUC scores. Eventually, as previously explained the best optimization results could start to be observed with four classifiers in space 2 when the proposed algorithm used the weights coefficient matrix scaled

at 0.01 as a function of the 1EXP (-) Z^+ search space reference $Z^+ = 2$, to observe the WVE combination scoring AUCs above 80% for all classes, with 83% AUC score for the macro average and 92% micro AUC score, which is 6% higher than the previous RF, SV, and KNN combinations micro AUC score of 86%.

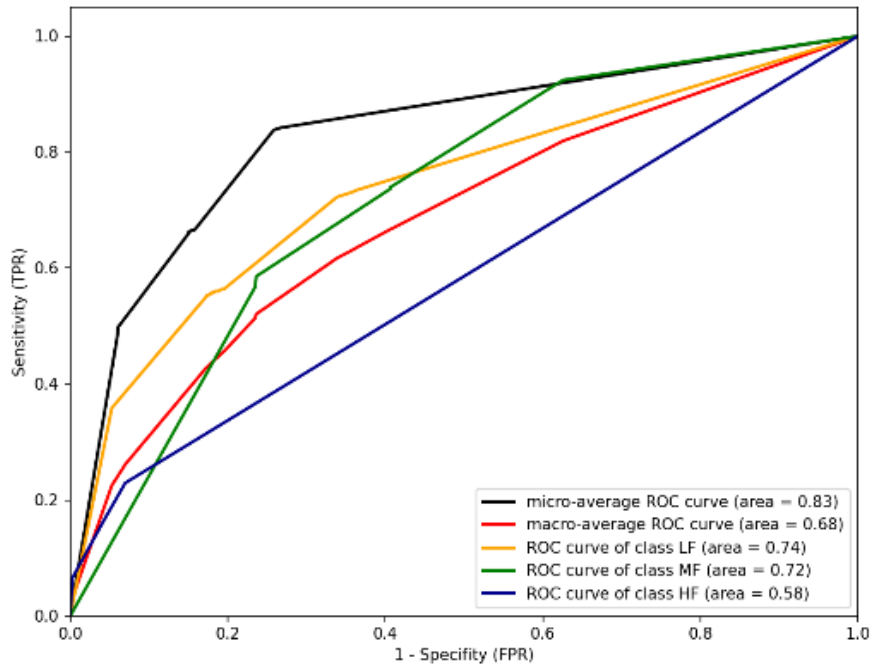


Figure 58: The DT and KNN combination ROC plots and AUC scores

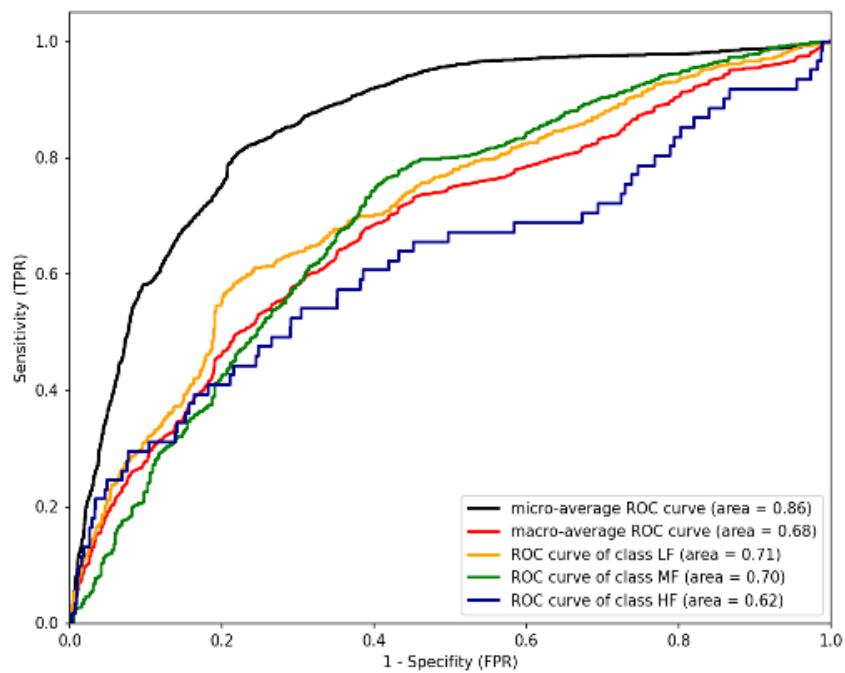


Figure 59: The RF, SV, and KNN combination ROC plots and AUC scores

The ROC plots for the best WVE combinations involving four and five classifiers probability predictions were obtained in spaces 2 and 3 based on the WVE's accuracy performances. Whereby, the GB, DT, RF, SVM, and KNN (Fig. 60), as well as GB, RF, SVM, and KNN WVE (Fig. 61) combinations were observed to exhibit an increasing ability towards the prediction of correct outcomes with respective micro and macro averages AUC scores above 90% and 80%, having above 80% AUC score in increasingly predicting correctly all the *low*, *medium*, and *high* fertility classes, with these AUC scores being better than those in Viscarra Rossel *et al.* (2010).

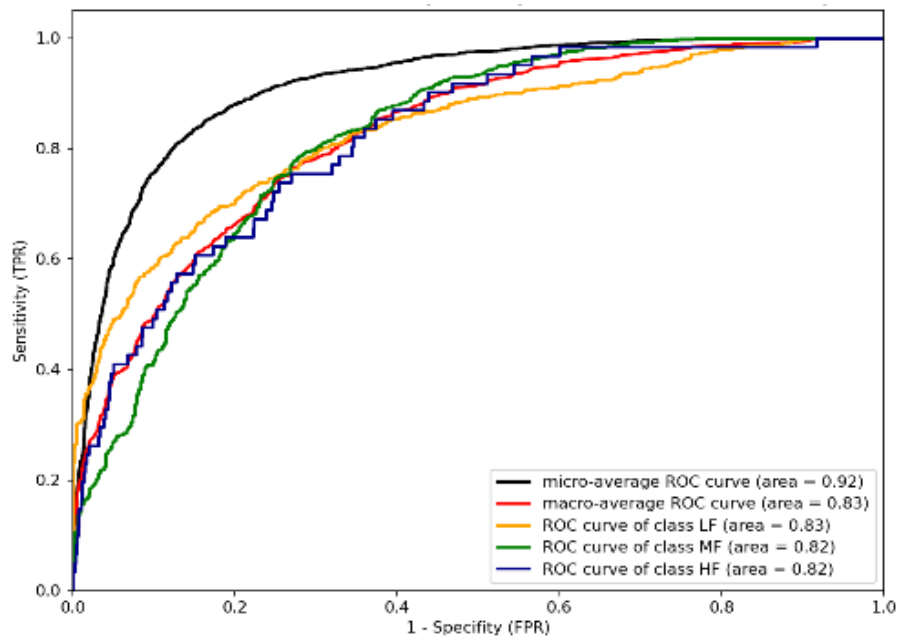


Figure 60: The GB, RF, SVM, and KNN combination ROC plots and AUC scores

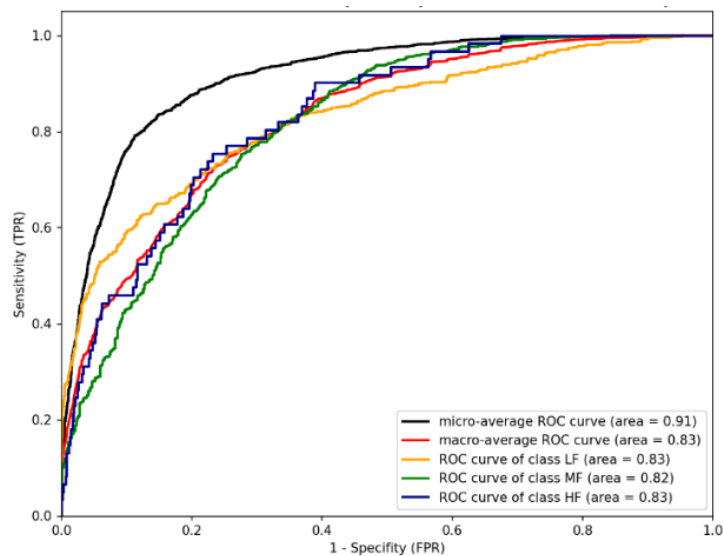


Figure 61: The GB, DT, RF, SVM, and KNN combination ROC plots and AUC scores

4.2.6 Comparative Analysis: The WVE vs Individual Models Results

Figure 62 shows the results of the Overall Optimal 2S-HHEC (that is the developed WVE) and individual models results. As compared to individual models, it could be observed that the WVE model could predict soil fertility status at performance higher than individual models, these results are as expected as it is the main goal of an ensemble to improve performance over its members.

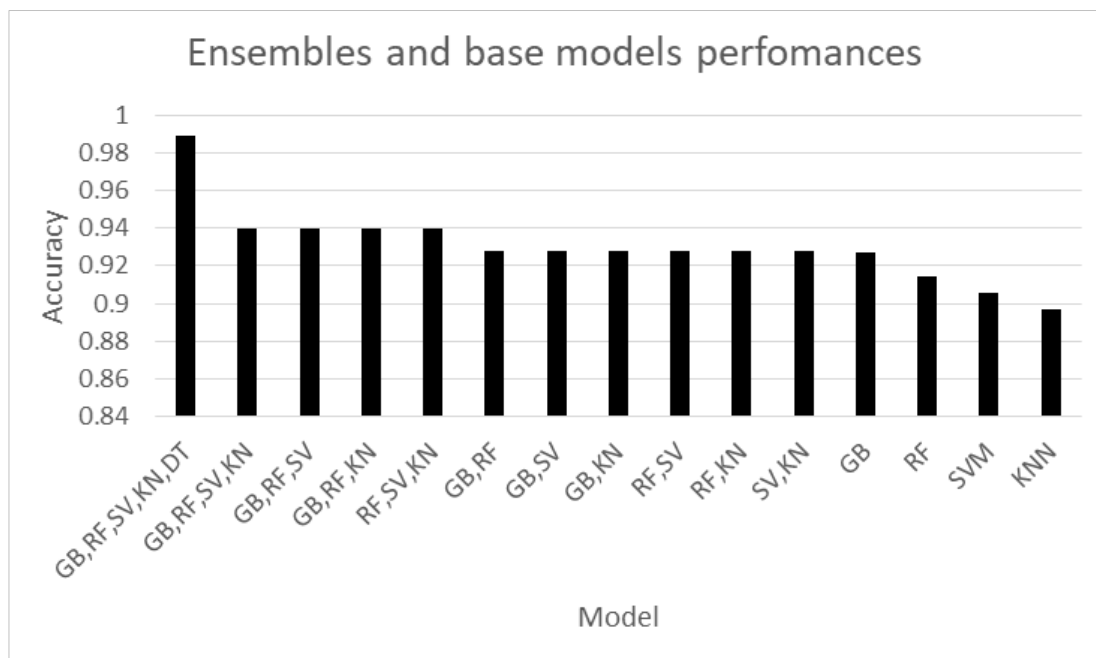


Figure 62: The 2S-HHEC's (WVE) and learners' performances

Table 15 provides a comparative analysis between the results of the new WVE model and benchmark model accuracy (in percentage) performance comparisons. It could be observed that GB, RF, SV, KN, and DT hybrid ensemble committee could achieve an accuracy of 98.93%, a score higher by approximately 1% than the result of authors in Jayalakshmi and Savitha (2022). However, it could be observed that, both the new model and benchmark had a Kappa score close to one, indicating that both models had a good interrater reliability indicating high agreement, although the benchmark has 1% higher as much agreement than the new model.

	New Model(s)		Benchmark(s)							
	Weighted voting		Bagging		Boosting		Stacking			
	GB,RF,SV,KN,DT	GB,RF,SV,KN	TreeBag	RF	C5.0	Gbm	LR	KNN	CRT	SVM
Accuracy	98.93	93.98	94.4	96.3	98.15	92.95	73.82	78.43	73.25	92.22
Kappa	93.9	87.2	88.06	92.24	94.9	76.98	48.01	30.58	19.5	79.36
AUC (Micro)	92	92	-	-	-	-	-	-	-	-
AUC (Macro)	83	83	-	-	-	-	-	-		-

Visual display of the model performances as compared to benchmark1 by Jayalakshmi and Savitha (2022) is shown in Fig. 63.

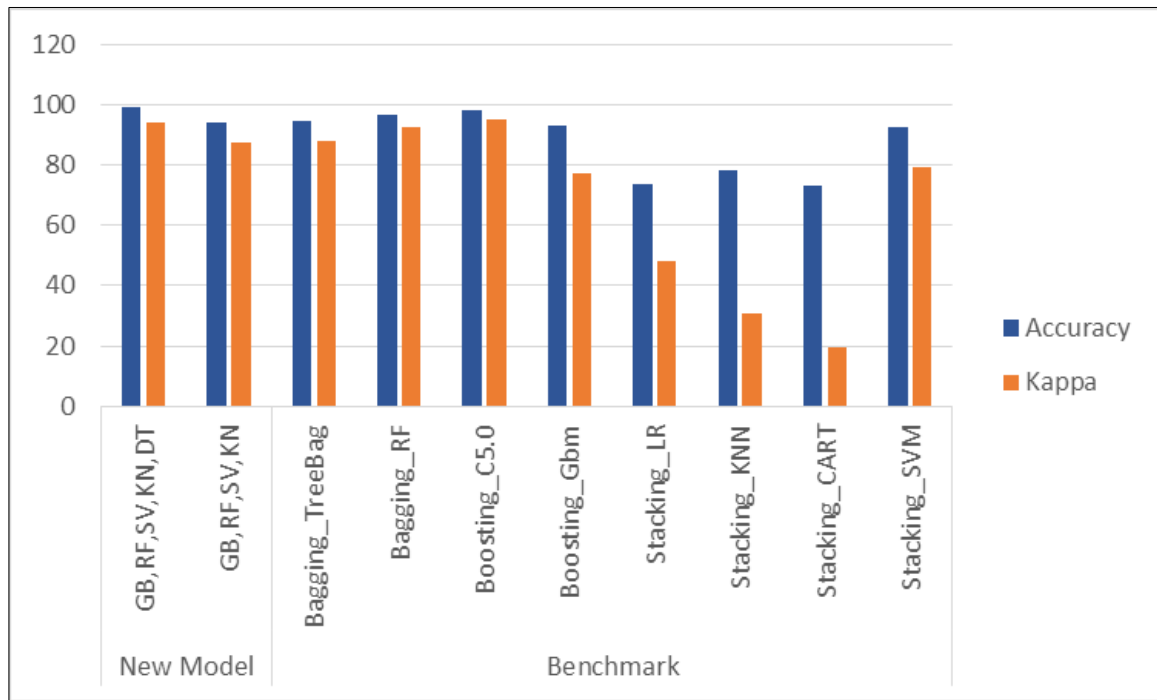


Figure 63: Visual display of the model performances as compared to benchmark 1

The ROC AUC of the new model in comparison with benchmark2 work by Rossel *et al.* (2010) are in displayed in Fig. 64. From that ROC analysis plot, it could be observed that the new model is above reasonably performing well with an increasing ability to provide correct predictions for the respective target soil fertility classes low, medium, and high classes, all showing to be above 0.5 cut point as shown in Fig. 64 which indicates that the model is extremely far from randomly guessing of predictions and performs reasonably well in correctly predicting each of those classes with actual AUC values of 0.87, 0.83 and 0.82.

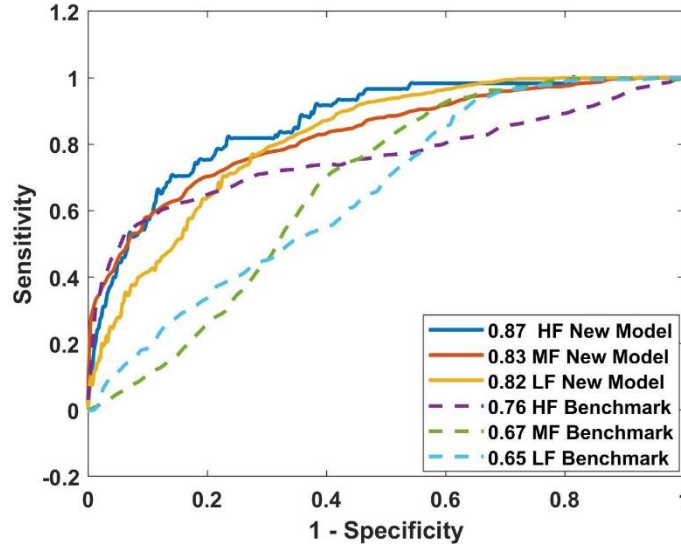


Figure 64: The ROC curves for the proposed model as compared to benchmark 2

These results had an improvement compared to the authors in Rossel *et al.* (2010). Additional results of the fertility target class predictions comparisons with benchmark results are shown in Figs. 65, 66, and 67 which displays the developed WVE model's ROC curves 0.87, 0.83, 0.82 for high, medium, and low fertility statuses or classes, the performance of which is better as compared to the benchmark2 model in Rossel *et al.* (2010) study which achieved ROC's curves of 0.76, 0.67, 0.65 for high, medium, and low fertility statuses or classes, respectively as shown in Figs. 66, 65, and 66. It could be observed the WVE model in this study had ROC very close to 1 as compared to results in the benchmark2 model. Successes in achieving such outstanding performance improvement was largely an effort of the automated weighting multi-precision values generation function which provides weighting values to be exhaustively searched for finding the optimal set.

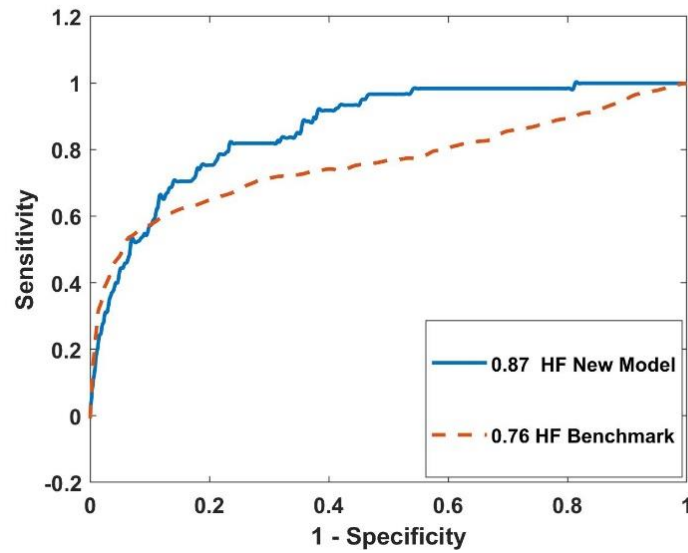


Figure 65: High fertility class prediction ROC curves for the proposed model as compared to benchmark 2

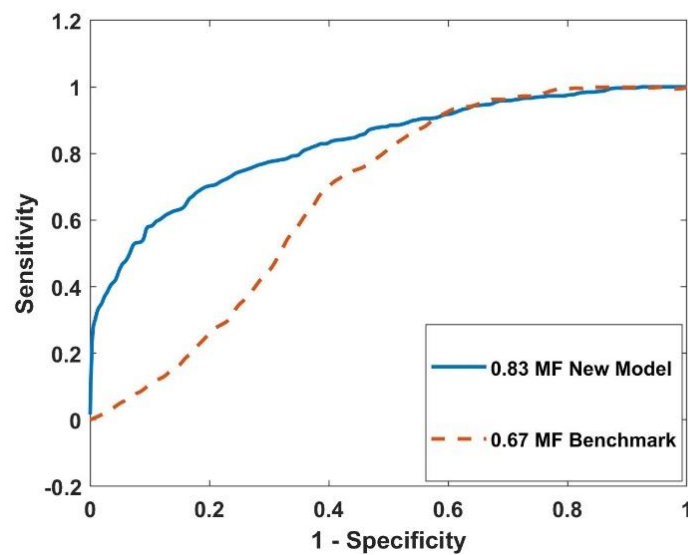


Figure 66: Medium Fertility Class Prediction ROC curves for the proposed model as compared to benchmark 2

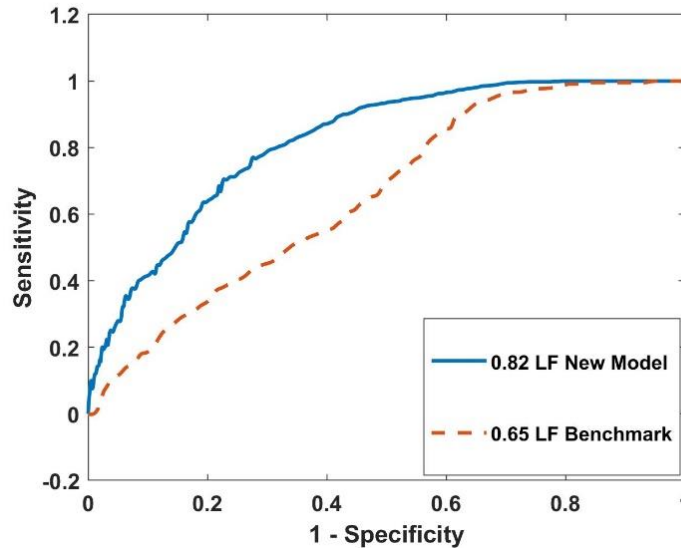


Figure 67: Low Fertility Class Prediction ROC curves for the proposed model as compared to benchmark 2

4.3 The WVE Model Validation and Utility Evaluation

The model was run with different unseen soil testing laboratory dataset of 62 samples to predict the samples' fertility status and validate the model against the laboratory-based fertility status test results as ground truth. Also, the model was run with another different 64 unseen samples that were collected from a maize plantation field to obtain predictions to use for recommendations of site-specific treatment wherever applicable. This was done before maize plantation experimentation to evaluate the utility and validity of the model predictions upon adherents to its corresponding or relevant recommendations in helping farmers increase crop yields, maize being the case study. These datasets were loaded in the model as batch files by using a Streamlit deployment-based interface that was developed in this research for the user-friendly accessibility of the model. Shown in Fig. 68 is the Streamlit-based interface for batch uploading soil properties data files into the model for prediction. The blue and yellow spheres depict the respective batch loading menu and input mode, and the obtained results are presented in sections 4.3.1. and 4.3.2.

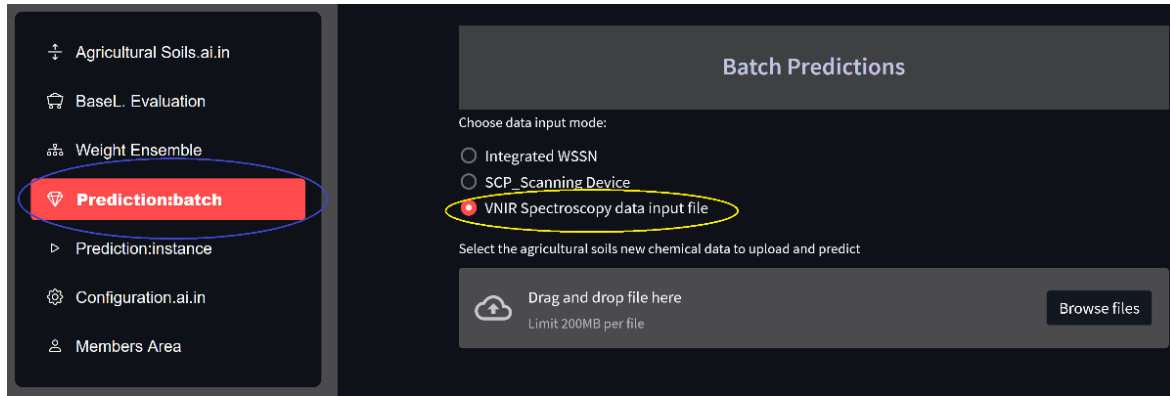


Figure 68: Streamlit-based interface for batch uploading soil properties data file into the model for prediction

4.3.1 Model Validation Results

Shown in Fig. 69 is a plot displaying the prediction result of just the WVE model (WVE_Preds) per sample, indicated in black dots, as well as those of laboratory results (LabTest) alone for the same sample, these are indicated by orange white-holed dots. Scenarios of correct predictions are indicated by the orange black-holed dots or donuts. Presented in the y-axis values of 0, 1, and 2 represent the respective low, medium, and high fertility statuses. It could be observed for sample number 1, both the model prediction that the sample had low soil fertility, as well as laboratory results read the same, leading to a correct classification, followed by samples 2 to 4 being classified correctly as medium. However, sample number 5 was classified as low by the VWE model while laboratory results read medium, leading to an incorrect classification. Samples number 6 to 16 were again correctly classified by the model, while sample 17 tested high in the laboratory, the model classified it as medium leading to an incorrect classification.

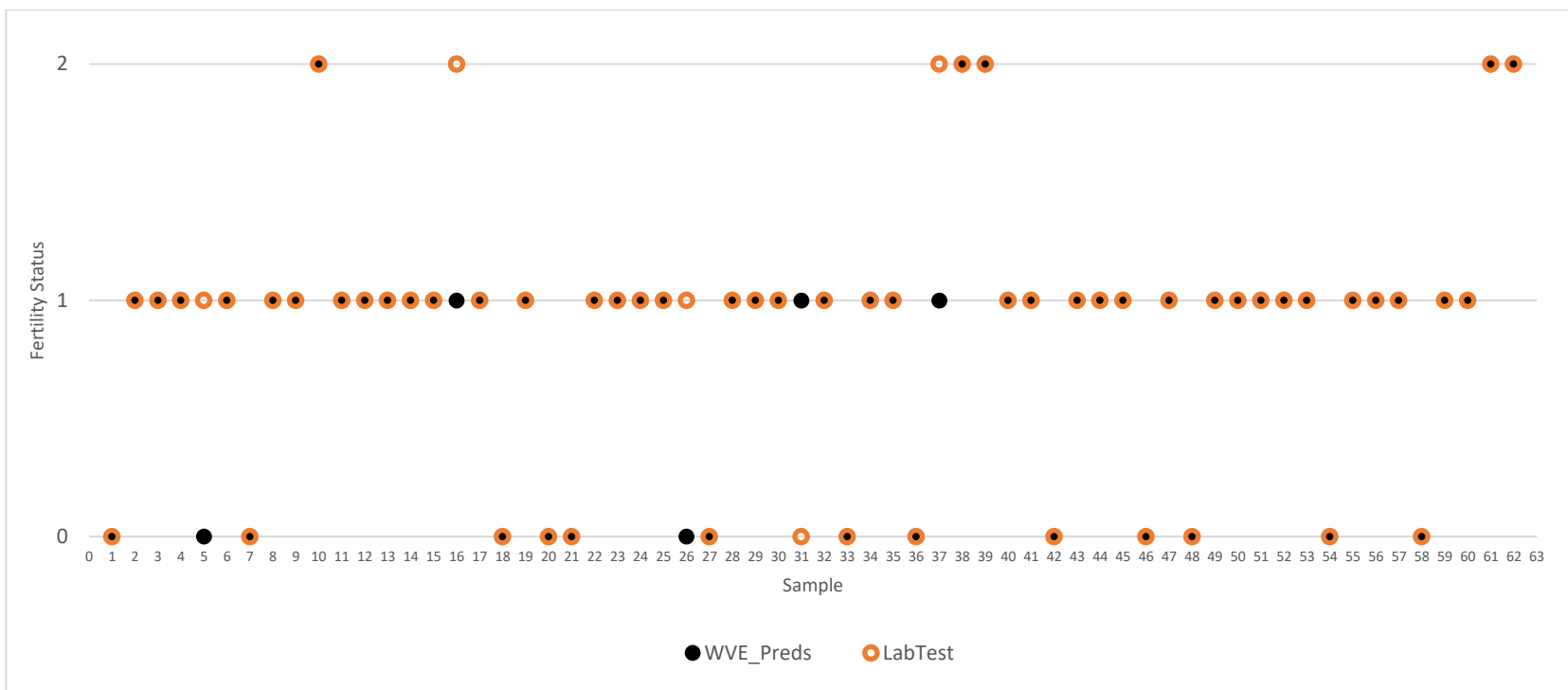


Figure 69: Plot of the WVE predictions vs Actual Soil Laboratory Test results

Eventually, out of the obtained 62 samples, 57 were correctly classified, and 5 were incorrectly classified by the WVE model, resulting in 92% correct classification, which is a fairly valid percentage with less than 10% of incorrect classification. In general, an overall characteristic of medium soil fertility status with dominance of 68% for Njombe, the soil had 24% characteristic of low fertility status, while it was inherent with high fertility status by just 8% (Figs. 70 and 71 of the respective percentage dominance in numeric and plotted).

Batch Samples Grand Summary and Visualization		
=====		
Low:	15	24%
Med:	42	68%
High:	5	8%

Figure 70: Summary of the soil fertility statuses WVE model predictions on Soil laboratory validation by numerical percentages

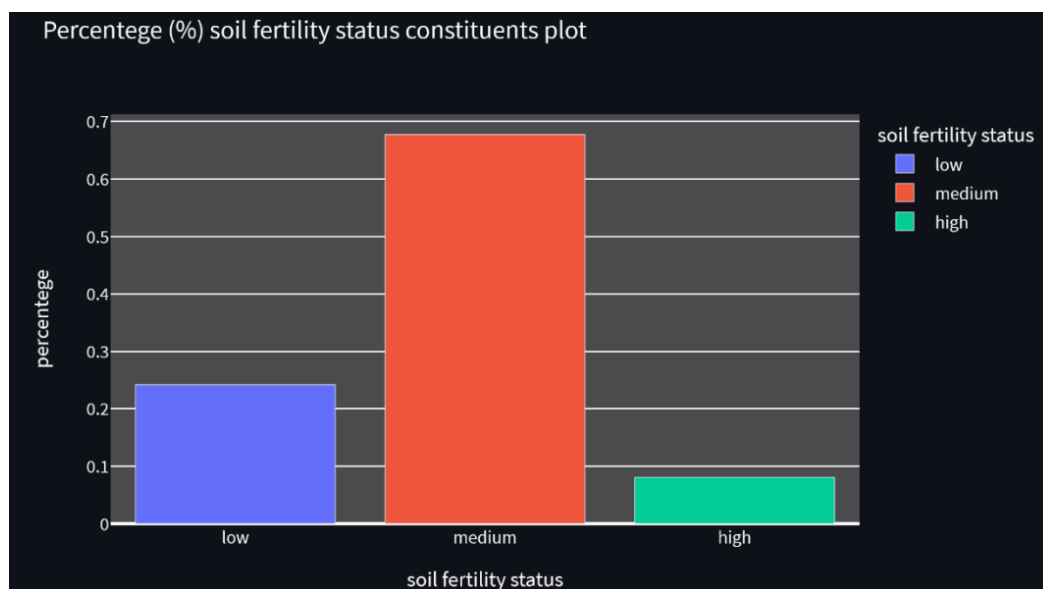


Figure 71: Plot of the Summary of soil fertility statuses predictions by WVE model using Soil laboratory validation dataset

4.3.2 Model Utility: WVE Model Predictions for Maize Plantations Field Grain Yields Experimentation Results

This section presents the results of field experimentation to validate the utility of the developed WVE in providing accurate predictions that are significant to support consequent soil fertility treatments and corresponding management practices as pre-plantation preparation. Whereby, fertility statuses for the 64 samples that were collected for the model-based section described were obtained following the proposed model prediction of the target fertility class group of measurement input values of constituting sample chemical composition. Figure 72 shows a plot of fertility status prediction results in percentage for the 64 batch samples of the model-based study section, where each prediction target of the sample was obtained for the 64 samples. As could be seen from Fig. 76 of the percentage-wise proportions of the predicted samples in the model-based study section, the samples were predicted approximately 47% lowly fertile and 42% medium fertile, and 11% high, the lowly fertility samples were highly treated using NPK fertilizer, and urea, and managed intensively throughout the experimentation process, those predicted medium fertility were moderately treated and managed, while those with high where not treated but only managed through appropriate monitoring until the harvest in each study section were obtained and weighted in tons for each of the section's 64 blocks.

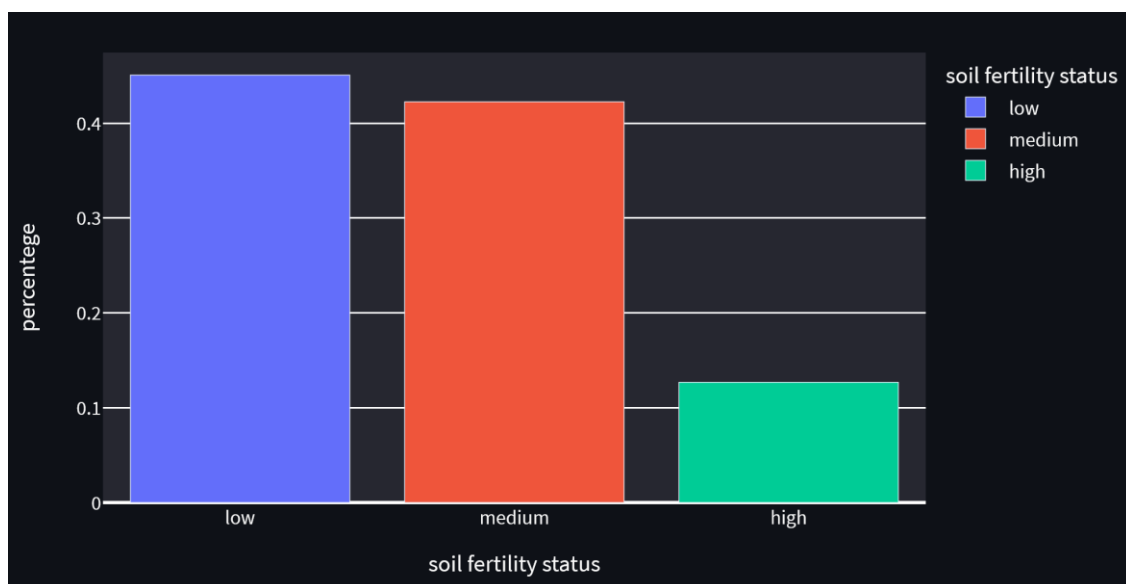


Figure 72: Percentage of soil fertility status constituents in the 64 model-based predictions section samples

As shown in Fig. 73 displays the harvest in tons for the 64 blocks for each study section. It could be seen that the model-based harvests were unnoticeably slightly higher as compared to a basic blanket, while as expected it was extremely higher than adhoc control based plantation harvests.

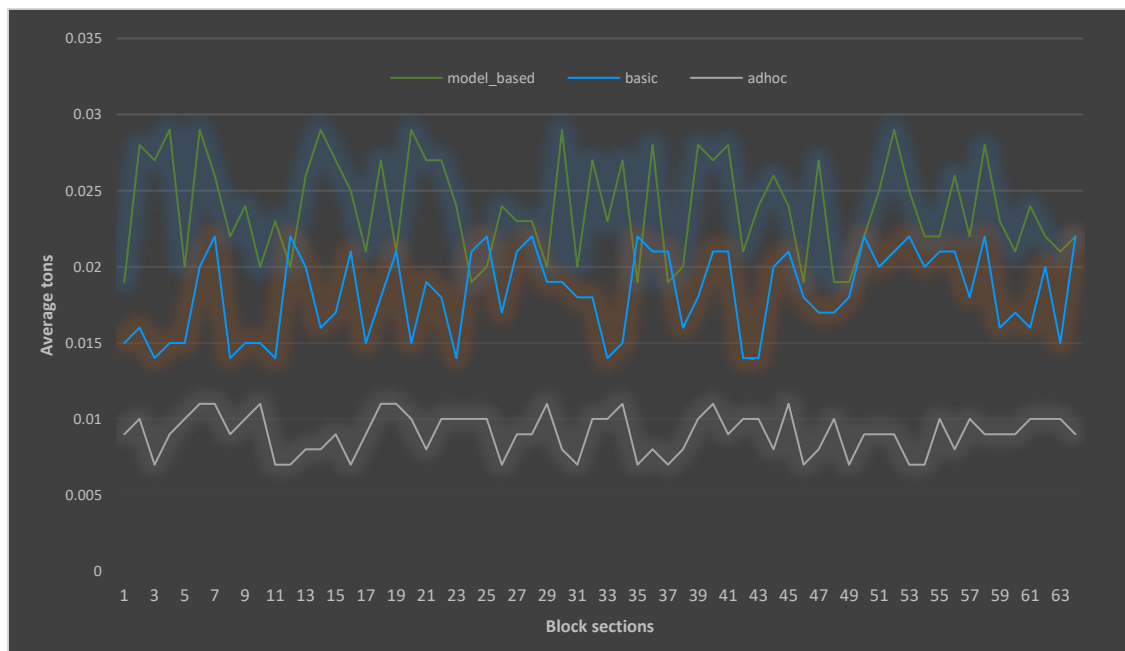


Figure 73: Plot of the percentages of predicted samples

Table 16 presents the total harvest tons amounts of maize per quarter by the study section, whereby, a total of 3.276 tons of maize were harvested in the 1-acre experimentation farmland. As compared to 4 tons per acre average harvest amounts for standard maize plantation experimentation, it could be noted that the total harvest of 3.276 tons is approximately 1 ton less and that could be explained by the fact that some of the deficit could have been due to the involvement of the Adhoc based soil fertility management results of 1.740 tons which must have significantly lowered the total yields in the conducted maize plantation experiment.

Table 15: Total harvest tons amounts of maize per quarter by study section

	1 st Qtr	2 nd Qtr	3 rd Qtr	4 th Qtr	<i>Total</i>	E.1A
Modal based	0.394	0.381	0.379	0.373	1.527	4.581
Adhoc control	0.143	0.150	0.145	0.142	0.580	1.740
Basic + blanket	0.271	0.297	0.290	0.311	1.169	3.507
Total	3.276					

As it can be seen from Fig. 74 of the percentage-wise proportions of the total harvest in each study section, harvest results of the model-based predictions information application were worthy, with a total of 1.527 tons harvested in the four-quarters of one-third of an acre, an amount which is 46% of the total amount harvested in that acre with the other remaining 54% being a constituent of the other non-model based section, 0.580 tons (18%) from adhoc controlled, and 1.169 tons (36%) from basic recommendations based.

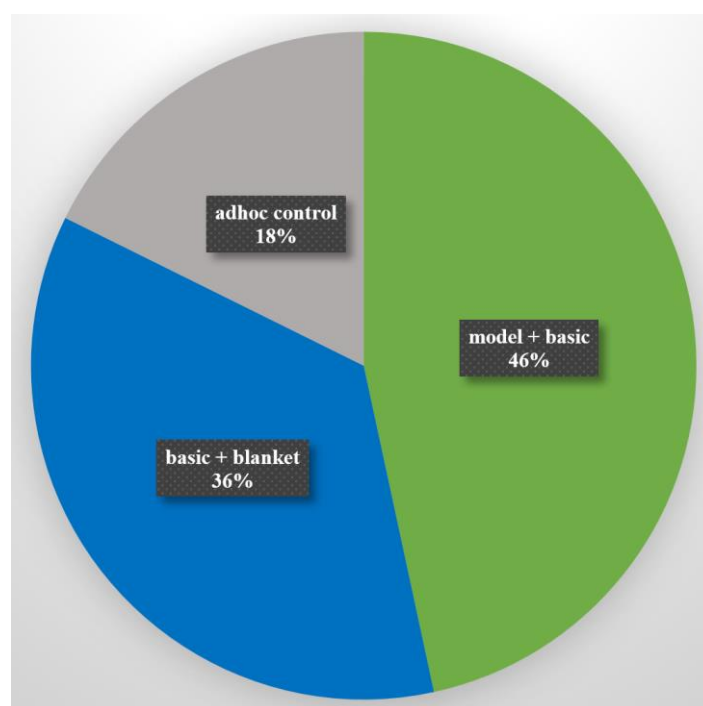
**Figure 74: Percentage-wise proportions of the total harvest in each study section**

Figure 75 displays the harvest in each study section, the total, and extrapolated to the 1-acre standard harvest amount in tons, i.e. for each one-third study section multiplied by 3 to form the extrapolated 1-acre harvest value (E.1A). Experimentations total maize grain yields or harvest results in tons per acre. As such, given everything remains constant and assuming the model-based procedures application across the remaining two-thirds of the farm to cover the

entire 1-acre farmland, we could assume to extrapolate the model-based harvest results in the one-third experimented onto an acre productivity estimate of 4.5 tons harvest (4.5 tons per acre) which is above the standard benchmark of average harvest amounts in similar maize plantation experimentation studies, and also it is above the average global maize production capacity of 4 tons per acre (Kipkulei *et al.*, 2022).

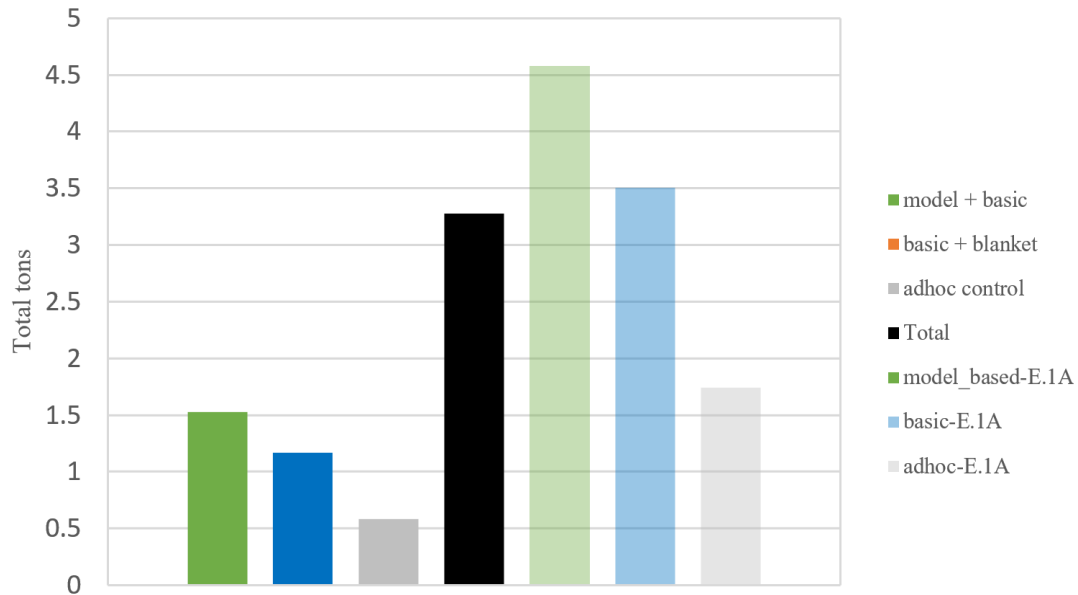


Figure 75: Harvest in each study section, totals, and extrapolation on an acre

4.4 Discussion

This research aimed at developing a high-performance soil fertility status prediction voting machine learning ensemble model using brute exhaustive optimization in automated 1EXP(-)Z⁺ multi-precision weights of hybrid classifiers for reliable prediction of agricultural soil fertility status using the optimal number of class targets. The ensemble was developed using a WVE scheme that combined individual base model class probability predictions using optimal weighting values that were found with brute optimization in novel 1EXP(-)Z⁺ based search spaces presented as weights coefficients matrices. In the beginning, while varying fertility statuses have been presented in previous studies, in this research an optimum number of three fertility targets low, medium, and high which are termed as classes were detected from soil chemical data by the automatic knee detection method. Positive test results were obtained following an analysis of variations (ANOVA) using the Tukey HSD test on the formed target clusters, which means the formed fertility clusters as targets are stably different. These of which were modeled by the K-means algorithm to obtain a labeled dataset which was consequently used as part of the overall heuristic to implement hybrid classifiers that were further combined

through an ensemble voting mechanism following weighted class probabilities prediction thereof. These results conform with the target classes used in previous studies by Chaudhari *et al.* (2020), Escorcia-Gutierrez *et al.* (2022), Kumar *et al.* (2019), and Rossel *et al.* (2010) who also used the same number of targets, that is three (3) classes to model classifiers for predicting soil fertility status. On the other hand, these results were different from Azhakarsamy and Sathiaselvan (2018), and Jayalakshmi and Savitha (2022) who used only 2 classes low and high. Also, the results were different from Bhuyar(2014) and Gholap *et al.* (2012) who used six (6) classes very low, low, medium/moderate, moderately high, high, and very high. Last but not least, these results also deviated from those by Manjula and Djodiltachoumy (2017) who used five (5) target classes very high, high, medium, low, and very low.

In addition, the brute exhaustive search-based WVE optimization procedure developed in this research finally came out with one robust ensemble model that could significantly improve the overall predictors' performance of the resultant solution model as a result of weighted voting the individual base learners' class probability predictions using appropriate weight coefficients for each base learner, to form the final weighted voting ensemble of heterogeneous hybrid data manipulation based hybrid predictors. In this study, the proposed resultant WVE model solution improved the soil fertility status predictive performance of the individual base model as expected. It could be observed that the obtained WVE model achieved a predictive accuracy of 98.93% which outperformed all of its base models whereby the maximum accuracy of its members was by gradient boosting hybrid with 93% accuracy. Also, as observed from the comparative results in Section 4, the proposed WVE model outperformed published benchmark model performances from previous studies. This is affirmed by comparing with one of the best soil fertility status predictive model performance results in the study by Jayalakshmi and Savitha (2022) which could achieve an accuracy of 98.15% which is approximately 1% less than the model in this study. Furthermore, the low, medium, and high ROC-AUC results of the WVE model developed in the study outperformed those in previous studies in predicting the different fertility class targets. For instance, the ROC-AUC model results in a study by Viscarra Rossel *et al.* (2010), attained areas under the curve of 65%, 67%, and 76%, for respective low, medium, and high target classes, which are less than those of the WVE results in this study that are 82%, 83%, and 87%, respectively. As such, this shows that the model solution presented in this study can reliably be used to predict soil fertility status at high performance for practical application with appreciable model operating characteristics. Last but not least The development herein, technical ascertaining as observed by the experimental results of the

model solution developed in this study suggested results that showed criticality of searching to find that optimal WVE with not only minding of the weights to be assigned to individual base experts but also in their diversities, such that the solutions subsets must comprehensively account for all possible ensemble diversities as it is being gradually expanded to involve additional expert(s) that may be of more significance in improving the VWE performance.

On the contrary, the resultant ensemble experts if selected subjectively without a fair elective search procedure that is designed to account for all other possibilities at hand, may end up with wrong results such as those observed with the VWE constituting of KNN and DT with respective weighs. Whereby we could see the resultant WVE ROC were sharply just around the 0.5 cut point, which signified that the ensemble was highly guessing as previously stated, and it is a wrong or bad model with probably a local optimal WVE configuration set to use with those particular prescribed base models with the prescribed weights. Unlike Jayalakshmi and Savitha (2022) who used homogenous ensembles, the use of a diverse ensemble base models was also suggested by Löfström (2015), Melville and Mooney (2003), and Rame and Cord (2021), where it was asserted that the use of heterogeneous classifiers to form an ensemble could be a best practice in attempts to capitalize on their strengths which may have significant effects in having better overall predictions as some base models could be weak in predicting particular classes while could be very strong in predicting other targets classes.

Furthermore, concerning the optimization technique, in this research, the results of the developed novel $1EXP(-)Z^+$ based arithmetic sequences multi-precision weights coefficients values matrix formulation algorithm were noticeably promising, the algorithm did manage to execute at a reasonable hardware computational complexity whereby it utilizes a fair discharge of memory and computational time, in turn, to provide arithmetic sequences based values using a WVE weights domain constraint asymptotic arithmetic series closed loop function, such an exploration which is previously nowhere to be seen. The brute exhaustive search finally used the proposed algorithm's function-generated values in the form of weights coefficients as search spaces to execute the brute search procedure to find the optimal WVE that was the most favorable soil fertility status prediction model with high accuracy performance.

The proposed VWE model provided an outstanding percentage correct classification of 92% on soil laboratory-based dataset of known sample's fertility status test results from the Njombe region. A general inherent medium soil fertility status was observed from the soil samples, with 68% dominance, 24% characteristic of low fertility status, and just 8% high fertility status

characterized. This does not conform to the majority of the fertility characteristic landscape previously observed in this research for the general modeled Tanzania soil data herein this study, whereby results indicated to be lowly fertility characterized. Therefore, these different Njombe results entail some of the regions do not have low soil fertility status, as it could also be seen from Mbeya's results, most of the landscape is highly fertility characterized. Finally, but not least, the proposed WVE model predictions were found to be useful for providing soil fertility status predictions which could assist in decision-making on site-specific appropriate remedies and management practices to apply in different farm field blocks as it could accurately predict soil fertility status, this predictive information of which could later on be valuable in the determination of locations with low and medium fertilities statuses suggesting that high treatment and subtle management practices need be applied in the former while moderately they should be applied in the later with medium fertility statuses. Such practices were similarly suggested by Gholap *et al.* (2012) and Manjula and Djodiltachoumy (2017). Eventually, this could highly make it possible to obtain improved maize grain yields productivity as shown in Fig. 76 for the comparison of the extrapolated 1-acre maize harvest value with the Tanzania 2019-2020 Agricultural year in tons per acre, in Tanzania Mainland.

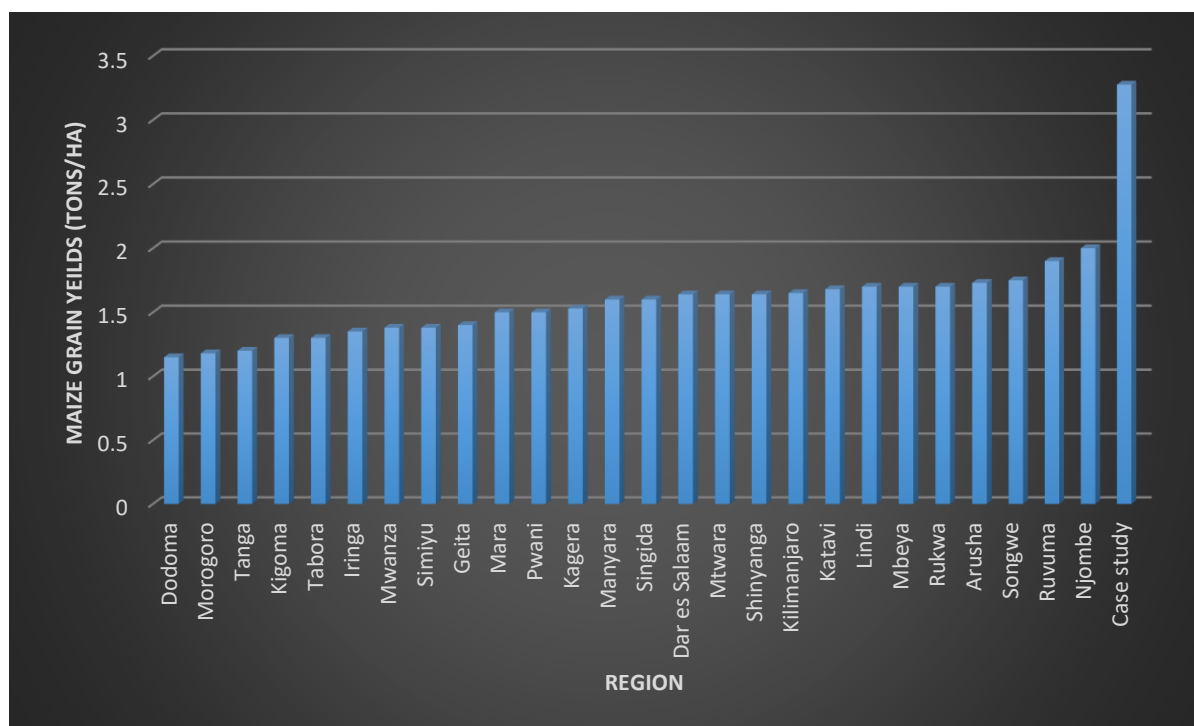


Figure 76: Comparison of the extrapolated 1-acre maize harvest value with the Tanzania 2019-2020 Agricultural year in tons per acre, in Tanzania Mainland

Generally, the improved maize grain yield results obtained from the study's field experimentation suggest that the model could highly be used to improve food productivity to ensure food security. Whereby more cohesion could be achieved through the proposed model implementation as part of the global's smart food production system.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this research, a high-performance soil fertility status prediction was developed and evaluated by using an optimal number of class targets through a weighted voting ensemble model that was developed through brute optimization in a novel weights domain constrained optimal $1EX(-)Z^+$ initial term-based arithmetic sequences initialization function for generating multi-precision weights coefficient matrices as search spaces. The proposed novel $1EXP(-)Z^+$ initial term-based arithmetic sequences multi-precision search spaces generation algorithm exhibited a mathematical validity to the WVE weights domain constraints algorithm as it portrayed an infinity with affinity asymptotic optimality within its boundaries. The search spaces formulation algorithm could effectively formulate multi-precision search spaces which could be used as part of a brute exhaustive search procedure to find appropriate weights configurations sets for soil fertility status prediction WVE. Whereby, ninety-four million (94 000 000) possible values were formulated in the stable search space 2 whose sequence initial term value is 0.01, with 100 values as search space weights points. Later, by using four (4) base models and the WVE weights domain constraints, these could be reduced to one hundred thirty-three thousand nine hundred and ninety-two (133 192) candidate base members vs weights combinations where the GB, RF, SVM, and KNN optimal WVE solution could be found at 94% prediction accuracy, 83% macro AUC average score, and 92% micro AUC average which was 6% higher than a previously obtained RF, SV and KN WVE combination at micro AUC score of 86%. Nevertheless, due to massive computational requirements that prematurely halt execution in search space 3 using Core i8 hardware with 64 GB RAM, with independent processing of the partially logged combinations to find a combination of weighted voting GB, DT, RF, SVM, and KNN classifiers ensemble with an accuracy score of 98.93% and Cohen Kappa 93.9% on test data was found to be the best alternative. This model could reliably predict each target class at respective 87%, 83%, and 82% for high, medium, and low fertility classes ROC-AUC scores.

In addition, field experimentation results showed that the model-based application could results in high maize yields with 1.5 tons in one-third of an acre, with a probable estimate of 4.5 tons in an acre, which is a harvest amount above the global maize production per acre, such that we

term it that our proposed model solution achieved both laboratory-based test simulation results as well as in the significance of its model predictions towards providing useful accurate optimal information about soil fertility status of various agricultural field block, in such the right interventions related to the provision of required remedy dosages were it applies could be made before plantation, and the observable results of the maize grain yields as a case study were noticeably good as previously demonstrated.

The research findings are summarized as follows:

- (i) The optimum number of soil fertility target classes, which were observed to have a high distribution of low soil fertility characteristics (s), these being a fact also previously coined in several studies about the situation of soil fertility across agricultural lands in Africa being at stake, such that this suggests that soil treatment before plantation and appropriate management needs be performed per a definite distanced farm fields proximity sites following a thorough analysis such as the one we presented in our solution.
- (ii) A design for modeling soil nutrients modeling through heterogamous hybrid WVE for the reliable prediction of soil fertility at high accuracy and ROC performances, as described in Section 4, is crucial in the implementation of a reliable soil fertility status ML model. The implementation of this design of which presented very outstanding results of performances higher than key benchmark published results of the same ML tasks.
- (iii) It was also found that the $1EXP(-) Z^+$ function is asymptotic to the WVE weights domain constraints. As such it can well be used to effectively implement an automated algorithmic procedure for WVE optimization with attainable computational complexity with both asymptotic analysis and H/W clock cycles by using algebra arithmetic sequences and matrices to represent the search spaces as variable weights coefficients values.
- (iv) The study also found that search spaces that are to be comprehensively searched through to find optimal configurations set are indeed very significant to the overall optimization of the weighted voting ensemble model using brute exhaustive search technique, let alone the diversities of the involved members. As such the fact stays that these spaces as well as heterogeneity are amongst the factors that highly may lead to

high-performing ensemble models through WVE implementation. As it could be seen a homogenous set of C.50 model was underscored by our proposed heterogeneous WVE model that was optimized in well-thought-of search spaces.

- (v) Likely, as the research implemented a generalized WVE model, it could be observed how at certain individual configurations thresholds could drop the performances upon analyzing the candidates' ROC we could observe a highly guessing and WRONG model. This might suggest versioning to a more non-generalized WVE implementation which accounts for both the members' hyperparameters and the WVE weights themselves.

5.2 Recommendations

As a recommendation to the government and corresponding agricultural entities, to make better use of the proposed model solution and results in this research to achieve smart soil fertility management. The government can also develop policies for the use of the model as part of the agricultural inputs subsidization. In turn, all this may facilitate sustainable agricultural intensification through precision agricultural provision system model(s) for the determination of site-specific soil fertility status deficiencies. Especially in these regions of Africa where it was since long been declared by Smaling *et al.* (1997) to be highly at stake. Additionally, this will suffice as a good response to the united nations' demand for developing technological solutions for a smart global food production system to improve food productivity and ensure food security through sustainable and digital agriculture (Jin *et al.*, 2019; United Nations, 2023). While, the challenges encountered in the implementation of machine learning and data science initiatives include language and cultural impediments, insufficient financial resources, suboptimal internet connectivity, and restricted availability of reliable and all-encompassing data. To address these challenges, the government must allocate resources towards data collection, network enhancements, computing infrastructure, and the promotion of education and training to cultivate local experience. Furthermore, the presented weighted voting ensembles model could be deployed for use by soil testing laboratories and farmers for them to gain a better understanding of farm fertility conditions. This will be achieved through cost-effective and timely accessibility to accurate soil fertility status predictions, this which may lead to the application of both site-specific and fine-tuned fertilizer dosages, as well as the appropriate management interventions necessary to ensure sustainability in soil fertility management as a key to sustainable agricultural intensification for ensuring food security for

the world population growth, which is estimated to reach 8 billion people by 2030, and 10 billion by 2050 (FAO, 2018). Finally but not least, Data scientists and ML engineers that aim to apply the WVE ML models performance improvement technique can use the 1EX (-) Z^+ initial term-based arithmetic sequences initialization function-based brute search algorithm that is developed in this research, for generating multi-precision weights coefficient matrices as a benchmark to effectively optimize ML WVE prediction models from various other domain such as medicine, just with slight adjustments of the resultant models ROC analysis cut point implementations to address the sensitivity of the particular domain. Nevertheless, while the developed algorithm can be used to significantly improve models performance by guaranteeing optimal solution finding, it cannot be applied to optimize very large WVEs, instead it should only be used in the context of smaller sized ensembles optimization. And, in order to reduce computational times, the created combination can be logged into caches for re-use, while also considerations for their implementation in ML WVE models optimization for tasks such as prediction of crop diseases or even medical related drugs discovery can be a useful consideration.

Future work may be to:

- (i) To develop a soil fertility status prediction model that incorporates available weather and climatic parameters benchmark data to learn more uncertainties that may have inherent featured effects of soil fertility.
- (ii) And also to deploy the developed model through a cloud-based environment with support for the user's front-end graphical interface, such as streamlit application-based deployment, for use in real-time soil fertility status prediction and fertilizer recommendation applications by farmers and relevant government and agricultural agencies.
- (iii) To conduct experiments for improving the efficiency of the proposed algorithm. Possibly exploring the implementation of the algorithm in a quantum computation implementation to capitalize on the rich qubit structure magic qubit combination as a key to represent the massive information from the developed 1EXP (-) Z^+ based brute exhaustive search technique, of which might also set provisions for accommodating more base experts to have much larger WVEs. And also to apply the developed 1EXP (-) Z^+ based brute exhaustive search technique to optimize weighted voting ensemble

models for improving the performances of different ML classification tasks for other domain-specific problems such as medicine and finance.

REFERENCES

- Adamchuk, V. I., Hummel, J. W., Morgan, M. T., & Upadhyaya, S. K. (2004). On-the-go soil sensors for precision agriculture. *Computers and Electronics in Agriculture*, 44(1), 71–91. <https://doi.org/10.1016/j.compag.2004.03.002>.
- Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., & Honrao, V. (2013). Predicting students' performance using ID3 and C4. 5 classification algorithms. *ArXiv Preprint ArXiv:1310.2071*.
- Ali, I., Greifeneder, F., Stamenkovic, J., Neumann, M., & Notarnicola, C. (2015). Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data. *Remote Sensing*, 7(12), 16398–16421.
- Alin, A. (2010). Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3), 370–374.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- Ameri, S., Fard, M. J., Chinnam, R. B., & Reddy, C. K. (2016). Survival analysis based framework for early prediction of student dropouts. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 903–912.
- Anastassiou, G. A. (2022). *Generalized symmetrical sigmoid function activated neural network multivariate approximation*. submitted.
- Angulo, A., Rodríguez, D., Garzón, W., Gómez, D. F., Al Sumaiti, A., & Rivera, S. (2021). Algorithms for bidding strategies in local energy markets: Exhaustive search through parallel computing and metaheuristic optimization. *Algorithms*, 14(9), 269.
- Anifowose, F. (2020). *Hybrid Machine Learning Explained in Nontechnical Terms*. JPT. <https://jpt.spe.org/hybrid-machine-learning-explained-nontechnical-terms>.
- Anifowose, F. A., Labadin, J., & Abdulraheem, A. (2013). Prediction of petroleum reservoir properties using different versions of adaptive neuro-fuzzy inference system hybrid models. *International Journal of Computer Information Systems and Industrial Management Applications*, 5, 413–426.

- Ariyanti, I., Ganiardi, M. A., & Oktari, U. (2019). Mobile Application Searching of the Shortest Route on Delivery Order of CV. Alfa Fresh With Brute Force Algorithm. *Logic: Jurnal Rancang Bangun Dan Teknologi*, 19(3), 120–130.
- Arooj, A., Riaz, M., & Akram, M. N. (2018). Evaluation of predictive data mining algorithms in soil data classification for optimized crop recommendation. *2018 International Conference on Advancements in Computational Sciences*, 1–6.
- Ast, J., Wasseghi, R., & Nyhuis, P. (2021). A comparison of methods for determining performance based employee deployment in production systems. *Production Engineering*, 15(3), 335–342.
- Ayodele, T. O. (2010). Types of machine learning algorithms. In *New advances in machine learning*. IntechOpen.
- Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An introduction to machine learning. *Clinical Pharmacology & Therapeutics*, 107(4), 871–885.
- Bagheri, B.M., Moghimi, A., Navidi, M. N., & Ebrahimi, M. F. (2018). Modeling of yield and rating of land characteristics for corn based on artificial neural network and regression models in southern Iran. *Desert*, 23(1), 85–95.
- Ball, B. C., Guimarães, R. M., Cloy, J. M., Hargreaves, P. R., Shepherd, T. G., & McKenzie, B. M. (2017). Visual soil evaluation: A summary of some applications and potential developments for agriculture. *Soil and Tillage Research*, 173, 114–124.
- Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. *Journal of the Association for Information Systems*, 19(5), 3.
- Baskerville, R., Pries-Heje, J., & Venable, J. (2009). Soft design science methodology. *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, 1–11.
- Baştanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. *MiRNomics: MicroRNA Biology and Computational Analysis*, 105–128.

- Bellman, R. (1997). *Introduction to matrix analysis*. SIAM.
- Bhattacharya, B., & Solomatine, D. P. (2006). Machine learning in soil classification. *Neural Networks*, 19(2), 186–195.
- bhspencer. (2015). *Answer to “How is Greedy Technique different from Exhaustive Search?”* Stack Overflow. <https://stackoverflow.com/a/31234596/19238581>.
- Bhuyar, V. (2014). Comparative Analysis of classification techniques on soil data to predict fertility rate for Aurangabad district. *International Journal of Emerging Trends Technology in Computer Science*, 3(2), 200–203.
- Bonaccorso, G. (2017). *Machine Learning Algorithms*. Packt Publishing Ltd.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Brownlee, J. (2016). *Data Mining with Weka*.
- Brownlee, J. (2018). *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery.
- Brownlee, J. (2020a). Combined Algorithm Selection and Hyperparameter Optimization. *Machine Learning Mastery*. <https://machinelearningmastery.com/combined-algorithm-selection-and-hyperparameter-optimization/>.
- Brownlee, J. (2020b). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. *Machine Learning Mastery*. <https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>.
- Brownlee, J. (2021). How to Develop a Weighted Average Ensemble With Python. *Machine Learning Mastery*. <https://machinelearningmastery.com/weighted-average-ensemble-with-python/>.
- Bünemann, E. K., Bongiorno, G., Bai, Z., Creamer, R. E., De Deyn, G., de Goede, R., Fleskens, L., Geissen, V., Kuyper, T. W., & Mäder, P. (2018). Soil quality—A critical review. *Soil Biology and Biochemistry*, 120, 105–125.

- Burbidge, R., Trotter, M., Buxton, B., & Holden, S. (2001). Drug design by machine learning: Support vector machines for pharmaceutical data analysis. *Computers & Chemistry*, 26(1), 5–14.
- Burns, N., & Grove, S. K. (2010). *Understanding nursing research-eBook: Building an evidence-based practice*. Elsevier Health Sciences.
- Carter, J. V., Pan, J., Rai, S. N., & Galandiuk, S. (2016). ROC-ing along: Evaluation and interpretation of receiver operating characteristic curves. *Surgery*, 159(6), 1638–1645. <https://doi.org/10.1016/j.surg.2015.12.029>.
- Castle, N. (2017). *An Introduction to Machine Learning Algorithms*. Oracle + Datascience.
- Chan, K. R. (2022, June 24). How to test normality, skewness and kurtosis using Python. *Omics Diary*. <https://medium.com/omics-diary/how-to-test-normality-skewness-and-kurtosis-using-python-18fb2d8e35b9>.
- Chaudhari, R., Chaudhari, S., Shaikh, A., Chiloba, R., & Khadtare, T. D. (2020). Soil Fertility Prediction Using Data Mining Techniques. *Bulletin Monumental*, 21(01), 8.
- Chaurasia, V., & Pal, S. (2017). *Performance analysis of data mining algorithms for diagnosis and prediction of heart and breast cancer disease*.
- Checkland, P. (1981). *Systems thinking, systems practice* John Wiley & Sons. New York.
- CIA Factbook. (2018a). *GDP - composition by sector*. <https://www.indexmundi.com/factbook/fields/gdp-composition-by-sector>.
- CIA Factbook. (2018b). *Labor force—By occupation*. <https://www.indexmundi.com/factbook/fields/labor-force-by-occupation>.
- Committee, M. S. (2019). *754-2019-IEEE standard for floating-point arithmetic*.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>.
- Dauzhenka, T., Kundrotas, P. J., & Vakser, I. A. (2018). Computational Feasibility of an Exhaustive Search of Side-Chain Conformations in Protein-Protein Docking. *Journal of Computational Chemistry*, 39(24), 2012–2021.

- De-Arteaga, M., Herlands, W., Neill, D. B., & Dubrawski, A. (2018). Machine Learning for the Developing World. *ACM Transactions on Management Information Systems*, 9(2), 1–14. <https://doi.org/10.1145/3210548>.
- Deng, H., Zhou, Y., Wang, L., & Zhang, C. (2021). Ensemble learning for the early prediction of neonatal jaundice with genetic features. *BMC Medical Informatics and Decision Making*, 21, 1–11.
- Devi, M. P. K., Anthiyur, U., & Shenbagavadivu, M. S. (2016). Enhanced crop yield prediction and soil data analysis using data mining. *International Journal of Modern Computer Science*, 4(6).
- Dolzhiikova, I., Abibullaev, B., Sameni, R., & Zollanvari, A. (2021). An Ensemble CNN for Subject-Independent Classification of Motor Imagery-based EEG. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 319–324.
- Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10), 78–87.
- Durairaj, M., & Vijitha, C. (2014). Educational data mining for prediction of student performance using clustering algorithms. *International Journal of Computer Science and Information Technologies*, 5(4), 5987–5991.
- Ekbal, A., & Saha, S. (2011). Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach. *ACM Transactions on Asian Language Information Processing*, 10(2), 1–37.
- Emerson, S., Cranston, R. E., & Liss, P. S. (1979). Redox species in a reducing fjord: Equilibrium and kinetic considerations. *Deep Sea Research Part A. Oceanographic Research Papers*, 26(8), 859–878.
- Emmet-Booth, J. P., Forristal, P. D., Fenton, O., Ball, B. C., & Holden, N. M. (2016). A review of visual soil evaluation techniques for soil structure. *Soil Use and Management*, 32(4), 623–634.

- Ennoui, A., O.Sihamman, N., Sabri, A., & Aara, A. (2021). *A Weighted Voting Deep Learning Approach for Plant Disease Classification*.
- Escorcia-Gutierrez, J., Gamarra, M., Soto-Diaz, R., Pérez, M., Madera, N., & Mansour, F. R. (2022). *Intelligent Agricultural Modelling of Soil Nutrients and pH Classification Using Ensemble Deep Learning Techniques*.
- FAO. (2016). *Agricultural Outlook 2016-2025*. Paris: OECD Publishing.
- FAO. (2017a). *The future of food and agriculture – Trends and challenges*. Food and Agriculture Organization of the United Nations.
- FAO. (2017b). *The State of Food and Agriculture: Leveraging Food Systems for Inclusive Rural Transformation*. Food and Agriculture Organization of the United Nations.
- FAO. (2018). *Future of Food and Agriculture 2018: Alternative pathways to 2050*. FOOD & AGRICULTURE ORG.
- FAO. (2022). Agricultural Production Statistics. 2000–2020. In *FAOSTAT analytical brief series no. 41*. Rome, Italy. ISSN 2709-0078 [Online]. <https://www.fao.org/3/cb9180en/cb9180en.pdf>
- Ferrández-Pastor, F. J., García-Chamizo, J. M., Nieto-Hidalgo, M., & Mora-Martínez, J. (2018). Precision agriculture design method using a distributed computing architecture on internet of things context. *Sensors*, 18(6), 1731.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367–378.
- Friedman, J. H., Bentley, J. L., & Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3), 209–226.
- Garcia, L., & Quek, F. (1997). Qualitative research in information systems: Time to be subjective? *Information Systems and Qualitative Research: Proceedings of the IFIP*

TC8 WG 8.2 *International Conference on Information Systems and Qualitative Research, 31st May–3rd June 1997, Philadelphia, Pennsylvania, USA*, 444–465.

- Gholap, J. (2012). Performance tuning of J48 Algorithm for prediction of soil fertility. *ArXiv Preprint ArXiv:1208.3943*.
- Gholap, J., Ingole, A., Gohil, J., Gargade, S., & Attar, V. (2012a). Soil data analysis using classification techniques and soil attribute prediction. *ArXiv Preprint ArXiv:1206.1557*.
- Gholap, J., Ingole, A., Gohil, J., Gargade, S., & Attar, V. (2012b). Soil data analysis using classification techniques and soil attribute prediction. *ArXiv Preprint ArXiv:1206.1557*.
- Giddens, A. (1986). *The constitution of society: Outline of the theory of structuration* (Vol. 349). Univ of California Press.
- Golge, E. (2016). Brief History of Machine Learning. *A Blog From a Human-Engineer-Being*. <http://www.erogol.com/brief-history-machine-learning>.
- Gomes, H. M., Barddal, J. P., Enembreck, F., & Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Computing Surveys*, 50(2), 1–36.
- Gregor, S., & Hevner, A. R. (2013a). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 337–355.
- Gregor, S., & Hevner, A. R. (2013b). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, 37(2).
- Hall, D., McCool, C., Dayoub, F., Sunderhauf, N., & Upcroft, B. (2015). Evaluation of features for leaf classification in challenging conditions. *2015 IEEE Winter Conference on Applications of Computer Vision*, 797–804.
- Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference On*, 363–366.
- Hanley, J. A. (1989). Receiver operating characteristic (ROC) methodology: The state of the art. *Crit Rev Diagn Imaging*, 29(3), 307–335.

- Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., & Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms: A review with a new dynamic approach. *Information*, 10(12), 390.
- Havlin, J. L., Tisdale, S. L., Nelson, W. L., & Beaton, J. D. (2016). *Soil fertility and fertilizers*. Pearson Education India.
- He, A., He, J., Kim, R., Like, D., & Yan, A. (2017). An ensemble-based approach for classification of high-resolution satellite imagery of the Amazon Basin. *2017 IEEE MIT Undergraduate Research Technology Conference*, 1–4.
- Hebb, D. O. (1949). *The organization of behaviour* New York. Wiley & Sons.
- Hengl, T., Mendes de Jesus, J., Heuvelink, G. B. M., Ruiperez Gonzalez, M., Kilibarda, M., Blagotić, A., Shangguan, W., Wright, M. N., Geng, X., Bauer-Marschallinger, B., Guevara, M. A., Vargas, R., MacMillan, R. A., Batjes, N. H., Leenaars, J. G. B., Ribeiro, E., Wheeler, I., Mantel, S., & Kempen, B. (2017). SoilGrids250m: Global gridded soil information based on machine learning. *Plos One*, 12(2), e0169748. <https://doi.org/10.1371/journal.pone.0169748>
- Hevner, A., & Chatterjee, S. (2010a). Design science research in information systems. In *Design Research in Information Systems*, 9–22. Springer.
- Hevner, A., & Chatterjee, S. (2010b). Design science research in information systems. In *Design Research in Information Systems*, 9–22. Springer.
- Hevner, A. R., March, S. T., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105.
- Ho, T. K. (1995). Random Decision Forests. *Proceedings of the Third International Conference on Document Analysis and Recognition*, 1, 278–282. <https://doi.org/10.1109/ICDAR.1995.598994>.
- Ho, Y.C., & Pepyne, D. L. (2002). Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115, 549–570.
- Hochreiter, S. (1991a). Investigations on dynamic neural networks. *Diploma, Technische Universität München*, 91(1).

- Hochreiter, S. (1991b). Studies on Dynamic Neural Networks [in German] Diploma thesis. *TU München*.
- Hurwitz, J., & Kirsch, D. (2018). *Machine Learning for Dummies* (IBM limited edition). John Wiley and Sons, Inc.
- Insozhan, N., & Parthasarathy, D. V. (2017). Evaluation and Management of Soil Fertility. *International Journal of Pure and Applied Mathematics*, 117(8), 11–15.
- Ishengoma, F., & Athuman, M. (2018). *Internet of things to improve agriculture in sub sahara Africa-a case study*.
- Jadhav, S. D., & Channe, H. P. (2016). Comparative study of K-NN, naive Bayes and decision tree classification techniques. *International Journal of Science and Research*, 5(1).
- Janvier, N., Arcade, N., Eric, N., & Jean, N. (2021). *Machine Learning based Soil Fertility Prediction*. 8(7), 5.
- Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity*, 41, 37–54.
- Jaschke, D., & Montangero, S. (2023). Is quantum computing green? An estimate for an energy-efficiency quantum advantage. *Quantum Science and Technology*, 8(2), 025001.
- Jayalakshmi, R., & Savitha, D. M. (2022). Mining Agricultural Data to Predict Soil Fertility Using Ensemble Boosting Algorithm. *International Journal of Information Communication Technologies and Human Development*, 14(1), 1–10.
- Jayaraman, P., Yavari, A., Georgakopoulos, D., Morshed, A., & Arkayd, Z. (2016). *Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt*.
- Jh, H. (1975). Adaptation in natural and artificial systems. *Ann Arbor*.
- Jiang, L., Cai, Z., Wang, D., & Zhang, H. (2012). Improving Tree augmented Naive Bayes for class probability estimation. *Knowledge-Based Systems*, 26, 239–245. <https://doi.org/10.1016/j.knosys.2011.08.010>.

- Jin, Z., Azzari, G., You, C., Di Tommaso, S., Aston, S., Burke, M., & Lobell, D. B. (2019). Smallholder maize area and yield mapping at national scales with Google Earth Engine. *Remote Sensing of Environment*, 228, 115–128.
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260.
- Kaderzhanov, M., Memon, S. A., Saurbayeva, A., & Kim, J. R. (2021). An Exhaustive Search Energy Optimization Method for Residential Building Envelope in Different Climatic Zones of Kazakhstan. *Buildings*, 11(12), 633.
- Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. X. (2017). A review on the practice of big data analysis in agriculture. *Computers and Electronics in Agriculture*, 143, 23–37.
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.
- Kavvadias, V., Papadopoulou, M., Vavoulidou, E., Theocharopoulos, S., Malliaraki, S., Agelaki, K., Koubouris, G., & Psarras, G. (2018). Effects of carbon inputs on chemical and microbial properties of soil in irrigated and rainfed olive groves. In *Soil Management and Climate Change*, 137–150. Elsevier.
- Keerthan, K. T. G., Shubha, C., & Sushma, S. A. (2019). Random forest algorithm for soil fertility prediction and grading using machine learning. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 1301–1304.
- Kipkulei, H. K., Bellingrath-Kimura, S. D., Lana, M., Ghazaryan, G., Baatz, R., Boitt, M., Chisanga, C. B., Rotich, B., & Sieber, S. (2022). Assessment of Maize Yield Response to Agricultural Management Strategies Using the DSSAT–CERES-Maize Model in Trans Nzoia County in Kenya. *International Journal of Plant Production*, 1–21.
- Kittappa, R. K. (1993). A representation of the solution of the nth order linear difference equation with variable coefficients. *Linear Algebra and Its Applications*, 193, 211–222.
- Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: A decision-tree hybrid. *KDD*, 96, 202–207.

- Kommineni, M., Perla, S., & Yedla, D. B. (2018). *A Survey of using Data Mining Techniques for Soil Fertility*.
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1), 3–24.
- Koza, J. R. (1994). *Genetic programming II: Automatic discovery of reusable programs*. MIT press.
- Krupicka, J., Sarec, P., & Novak, P. (2016). Measurement of electrical conductivity of fertilizer NPK 20-8-8. In *15th Internal Scientific Conference "Engineering for Rural Development", Jelgava, Latvia*.
- Kumar, A., & Kannathasan, N. (2011a). A survey on data mining and pattern recognition techniques for soil data mining. *IJCSI International Journal of Computer Science Issues*, 8(3).
- Kumar, A., & Kannathasan, N. (2011b). A survey on data mining and pattern recognition techniques for soil data mining. *International Journal of Computer Science Issues*, 8(3).
- Kurz, C. F., Maier, W., & Rink, C. (2020). A greedy stacking algorithm for model ensembling and domain weighting. *BMC Research Notes*, 13(1), 1–6.
- Larasati, A., Hajji, A. M., & Dwiastuti, A. (2019). The relationship between data skewness and accuracy of Artificial Neural Network predictive model. *IOP Conference Series: Materials Science and Engineering*, 523(1), 012070.
- Learned-Miller, E. G. (2014). Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.
- Li, D. (2021). Application of artificial intelligence and machine learning based on big data analysis in sustainable agriculture. *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, 71(9), 956–969.

- Li, D., Luo, L., Zhang, W., Liu, F., & Luo, F. (2016). A genetic algorithm-based weighted ensemble method for predicting transposon-derived piRNAs. *BMC Bioinformatics*, 17(1), 1–11.
- Liu, Y., Wang, H., Zhang, H., & Liber, K. (2016). A comprehensive support vector machine-based classification model for soil quality assessment. *Soil and Tillage Research*, 155, 19–26.
- Löfström, T. (2015). *On effectively creating ensembles of classifiers: Studies on creation strategies, diversity and predicting with confidence* [PhD Thesis]. Department of Computer and Systems Sciences, Stockholm University.
- Luenendonk, M. (2022, June 21). The Ultimate List of Machine Learning Statistics for 2023. *FounderJar*. <https://www.founderjar.com/machine-learning-statistics/>.
- Malato, G. (2022, November 8). *A practical introduction to the Shapiro-Wilk test for normality*. Medium. <https://towardsdatascience.com/a-practical-introduction-to-the-shapiro-wilk-test-for-normality-5675e52cee8f>.
- Manjula, E., & Djodiltachoumy, S. (2017). Data mining technique to analyze soil nutrients based on hybrid classification. *International Journal of Advanced Research in Computer Science*, 8(8).
- Manjula, E., & Djodiltachoumy, S. (2017b). Data mining technique to analyze soil nutrients based on hybrid classification. *International Journal of Advanced Research in Computer Science*, 8(8).
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266.
- Markoff, J. (1990). Business Technology; What's the Best Answer? It's Survival of the Fittest. *New York Times*. <https://www.nytimes.com/1990/08/29/business/business-technology-what-s-the-best-answer-it-s-survival-of-the-fittest.html>.
- Marr, B. (2016). A Short History of Machine Learning – Every Manager Should Read. *Forbes*. <https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/#2a1a75f9323f>.

- Masri, D., Woon, W. L., & Aung, Z. (2015). Soil property prediction: An extreme learning machine approach. *International Conference on Neural Information Processing*, 18–27.
- Massawe, B. H., Subburayalu, S. K., Kaaya, A. K., Winowiecki, L., & Slater, B. K. (2018). Mapping numerically classified soil taxa in Kilombero Valley, Tanzania using machine learning. *Geoderma*, 311, 143–148.
- McClish, D. K. (1989). Analyzing a portion of the ROC curve. *Medical Decision Making*, 9(3), 190–195.
- McQueen, R. J., Garner, S. R., Nevill-Manning, C. G., & Witten, I. H. (1995). Applying machine learning to agricultural data. *Computers and Electronics in Agriculture*, 12(4), 275–293.
- Mduma, N., Kalegele, K., & Machuve, D. (2019). A Survey of Machine Learning Approaches and Techniques for Student Dropout Prediction. *Data Science Journal*, 18(1).
- Melville, P., & Mooney, R. J. (2003). Constructing diverse classifier ensembles using artificial training examples. *International Joint Conference on Artificial Intelligence*, 3, 505–510.
- Melville, P., Shah, N., Mihalkova, L., & Mooney, R. J. (2004). Experiments on ensembles with missing and noisy data. *International Workshop on Multiple Classifier Systems*, 293–302.
- Menaga, A., & Vasantha, S. (2022). Smart Sustainable Agriculture Using Machine Learning and AI: A Review. *Ambient Communications and Computer Systems: Proceedings of RACCCS 2021*, 447–458.
- Michalski, R. S. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of development an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2), 125–161.

- Minh, D. H. T., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., & Maurel, P. (2017). Deep Recurrent Neural Networks for mapping winter vegetation quality coverage via multi-temporal SAR Sentinel-1. *ArXiv Preprint ArXiv:1708.03694*.
- Mishra, S., Mishra, D., & Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: A review paper. *Indian Journal of Science and Technology*, 9(38).
- Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
- Moore, A. (1991). Fast, robust adaptive control by learning only forward models. *Advances in Neural Information Processing Systems*, 4.
- Moore, A. W., Atkeson, C. G., & Schaal, S. A. (1995). *Memory-based learning for control*. Carnegie Mellon University, the Robotics Institute.
- Moore, A. W., Hill, D. J., & Johnson, M. P. (1992). An empirical investigation of brute force to choose features, smoothers and function approximators. *Computational Learning Theory and Natural Learning Systems*, 3, 361-379.
- Mouret, J. B., & Clune, J. (2015). Illuminating search spaces by mapping elites. *ArXiv Preprint ArXiv:1504.04909*.
- Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons*, 4, 51–62.
- Ndakidemi, P. A., & Semoka, J. M. R. (2006). Soil fertility survey in western Usambara Mountains, northern Tanzania. *Pedosphere*, 16(2), 237–244.
- Negied, N. K. (2014). Expert system for wheat yields protection in Egypt (ESWYP). *International Journal of Innovative Technology and Exploring Engineering*, 2278–3075.
- Nilsson, N. J. (1999). *Introduction to Machine Learning, Department of Computer Science*. Stanford University. <https://ai.stanford.edu/~nilsson/MLBOOK.pdf>.

- Nuankaew, W., Nuankaew, P., Doenribam, D., & Jareanpon, C. (2022). Weighted Voting Ensemble for Depressive Disorder Analysis with Multi-objective Optimization. *Current Applied Science and Technology*, 10–55003.
- Nyambo, D. G., Luhanga, E. T., & Yonah, Z. Q. (2019). A review of characterization approaches for smallholder farmers: Towards predictive farm typologies. *The Scientific World Journal*, 2019.
- Obuchowski, N. A., & Bullen, J. A. (2018). Receiver operating characteristic (ROC) curves: Review of methods with applications in diagnostic medicine. *Physics in Medicine & Biology*, 63(7), 07TR01.
- Okey, O. D., Maidin, S. S., Adasme, P., Lopes Rosa, R., Saadi, M., Carrillo Melgarejo, D., & Zegarra Rodríguez, D. (2022). BoostedEnML: Efficient Technique for Detecting Cyberattacks in IoT Systems Using Boosted Ensemble Machine Learning. *Sensors*, 22(19), 7409.
- Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research*, 2(1), 1–28.
- Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3), 128–138.
- Parr, T. (2018, June 27). *Answer to “Could you explain how gradient boosting algorithm works?”* Cross Validated. <https://stats.stackexchange.com/a/353467>.
- Partalas, I., Tsoumakas, G., & Vlahavas, I. P. (2008). Focused Ensemble Selection: A Diversity-Based Method for Greedy Ensemble Selection. *European Conference on Artificial Intelligence*, 117–121.
- Pastur-Romay, L. A., Cedrón, F., Pazos, A., & Porto-Pazos, A. B. (2016). Deep artificial neural networks and neuromorphic chips for big data analysis: Pharmaceutical and bioinformatics applications. *International Journal of Molecular Sciences*, 17(8), 1313.
- Pearce, J., & Ferrier, S. (2000). Evaluating the predictive performance of habitat models developed using logistic regression. *Ecological Modelling*, 133(3), 225–245.

- Pedamkar, P. (2019). Brute Force Algorithm | A Quick Glance of Brute Force Algorithm. *EDUCBA*. <https://www.educba.com/brute-force-algorithm/>.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- Peffer, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. *First International Conference on Design Science Research in Information Systems and Technology*, 83–16.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77.
- Pintelas, P. E., & Livieris, I. E. (2020). *Ensemble algorithms and their applications*. MDPI-Multidisciplinary Digital Publishing Institute.
- Poorinmohammad, N., Mohabatkar, H., Behbahani, M., & Biria, D. (2015). Computational prediction of anti HIV-1 peptides and in vitro evaluation of anti HIV-1 activity of HIV-1 P24-derived peptides. *Journal of Peptide Science*, 21(1), 10–16.
- Pospíšil, M. (2020). Representation of solutions of systems of linear differential equations with multiple delays and nonpermutable variable coefficients. *Mathematical Modelling and Analysis*, 25(2), 303–322.
- PyShark. (2021, July 25). *Skewness in Python*. PyShark. <https://pyshark.com/skewness-in-python/>.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106.
- Rajeswari, V., & Arunesh, K. (2016). Analysing soil data using data mining classification techniques. *Indian Journal of Science and Technology*, 9(19), 1–4.
- Ramane, D. V., Patil, S. S., & Shaligram, A. D. (2015). Detection of NPK nutrients of soil using Fiber Optic Sensor. *International Journal of Research in Advent Technology Special Issue National Conference ACGT 2015*, 13–14.

- Rame, A., & Cord, M. (2021). Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. *ArXiv Preprint ArXiv:2101.05544*.
- Raskulinec, G. M., & Fiksmen, E. (2015). SIMD Functions Via OpenMP. In *High Performance Parallelism Pearls* (pp. 421–440). Elsevier. <https://doi.org/10.1016/B978-0-12-803819-2.00006-9>
- Royston, J. P. (1983). Some techniques for assessing multivariate normality based on the Shapiro-Wilk W. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 32(2), 121–133.
- Royston, P. (1992). Approximating the Shapiro-Wilk W-test for non-normality. *Statistics and Computing*, 2, 117–119.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, 5(3), 1.
- Rurinda, J., Zingore, S., Jibrin, J. M., Balemi, T., Masuki, K., Andersson, J. A., Pampolino, M. F., Mohammed, I., Mutegi, J., & Kamara, A. Y. (2020). Science-based decision support for formulating crop fertilizer recommendations in sub-Saharan Africa. *Agricultural Systems*, 180, 102790.
- Sai, R., & Sathiaselan, J. G. R. (2018). Comparison of classifiers to predict classification accuracy for soil fertility. *International Journal of Advanced Studies of Scientific Research*, 3(9).
- Saini, H. S., Kamal, R., & Sharma, A. N. (2002). Web based fuzzy expert system for integrated pest management in soybean. *International Journal of Information Technology*, 8(1), 55–74.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210–229.
- Sathya, R., & Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2), 34–38.

- Shahhosseini, M., Hu, G., & Pham, H. (2019). Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *ArXiv Preprint ArXiv:1908.05287*.
- Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2020). Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access*, 9, 4843–4873.
- Sharma, A., Weindorf, D. C., Wang, D., & Chakraborty, S. (2015). Characterizing soils via portable X-ray fluorescence spectrometer: 4. Cation exchange capacity (CEC). *Geoderma*, 239, 130–134.
- Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., & Hsieh, C. J. (2017). Gradient boosted decision trees for high dimensional sparse output. *International Conference on Machine Learning*, 3182–3190.
- Siegelmann, H., & Sontag, E. (1995). Computational Power of Neural Networks. *Journal of Computer and System Sciences*, 50(1), 132–150.
- Simon. (2015, July 5). *Answer to “How is Greedy Technique different from Exhaustive Search?”* Stack Overflow. <https://stackoverflow.com/a/31234675/19238581>.
- Sinam, I., & Lawan, A. (2019). *An Improved C4.5 Model Classification Algorithm Based On Taylor’s Series*. Vol. 05, No. 01, April 2019, 9.
- Sirsat, M. S., Cernadas, E., Fernández-Delgado, M., & Barro, S. (2018). Automatic prediction of village-wise soil fertility for several nutrients in India using a wide range of regression methods. *Computers and Electronics in Agriculture*, 154, 120–133.
- Sirsat, M. S., Cernadas, E. G., & Delgado, M. F. (2017). *Application of machine learning to agricultural soil data* [PhD Thesis]. Universidade de Santiago de Compostela.
- Sirsat, M. S., Cernadas, E. G., Delgado, M. F., & Khan, R. (2017). Classification of agricultural soil parameters in India. *Computers and Electronics in Agriculture*, 135, 269–279.
- Smaling, E. M., Nandwa, S. M., & Janssen, B. H. (1997). Soil fertility in Africa is at stake. *Replenishing Soil Fertility in Africa*, 51, 47–61.

- Soleymani, R., Granger, E., & Fumera, G. (2020). F-measure curves: A tool to visualize classifier performance under imbalance. *Pattern Recognition*, 100, 107146.
- Suresh, S. (2018). *Nursing research and statistics*. Elsevier Health Sciences.
- Tan, P. N. (2007). *Introduction to data mining*. Pearson Education India.
- Taneja, S., Arora, R., & Kaur, S. (2012). Mining of soil data using unsupervised learning technique. *International Journal of Applied Engineering Research*, 7(11), 2012.
- Team, T. A. E. (2021, April 2). *Genetic Algorithm (GA) Introduction with Example Code*. Medium. <https://pub.towardsai.net/genetic-algorithm-ga-introduction-with-example-code-e59f9bc58eaf>.
- Turing. (2023). *How to Calculate Skewness and Kurtosis in Python?* <https://www.turing.com/kb/calculating-skewness-and-kurtosis-in-python>.
- UN. (2023). *Sustainable and Digital Agriculture | United Nations Development Programme*. <https://www.undp.org/policy-centre/singapore/sustainable-and-digital-agriculture>.
- Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: Innovating information and communication technology*. Crc Press.
- Vaishnavi, V., & Kuechler, W. (2004). *Design research in information systems*.
- Vaishnavi, V., & Kuechler, W. (2016). *Design Science Research in Information Systems, Association for Information Systems (2004)*.
- Vaishnavi, V., & Kuechler, W. (2017). *Design Research in Information Systems. AISWorld (2004)*.
- Vapnik, V., S, G., & A, S. (1997). *Support vector method for function approximation, regression estimation, and signal processing* (M. Mozer, M. Jordan, and T. Petsche). MIT Press.
- Varshney, P. (2020, October 17). *Q-Q Plots Explained*. Medium. <https://towardsdatascience.com/q-q-plots-explained-5aa8495426c0>.

- Venable, J. (2006). A framework for design science research activities. *Emerging Trends and Challenges in Information Technology Management: Proceedings of the 2006 Information Resource Management Association Conference*, 184–187.
- Rossel, R. A., Rizzo, R., Demattê, J. A. M., & Behrens, T. (2010a). Spatial Modeling of a Soil Fertility Index using Visible–Near-Infrared Spectra and Terrain Attributes. *Soil Science Society of America Journal*, 74(4), 1293–1300.
- Rossel, R. A., Rizzo, R., Demattê, J. A. M., & Behrens, T. (2010b). Spatial Modeling of a Soil Fertility Index using Visible–Near-Infrared Spectra and Terrain Attributes. *Soil Science Society of America Journal*, 74(4), 1293–1300.
- Walsh, M., Meliyo, J., Wu, W., Chen, J., Shepherd, K., Ekise, C., Simbila, W., Sila, A. M., Zhan, Y., & Mulvey, J. (2018). *Tanzania Soil Information Service (TanSIS)*. <https://doi.org/10.17605/OSF.IO/4NGAU>.
- Walter, A., Finger, R., Huber, R., & Buchmann, N. (2017). Opinion: Smart farming is key to developing sustainable agriculture. *Proceedings of the National Academy of Sciences*, 114(24), 6148–6150.
- Wang, J., Gao, B., & Yan, Y. (2022, September). Research on side channel attack based on bagging ensemble strategy. In *Third International Conference on Artificial Intelligence and Electromechanical Automation*, 12329, 648–656.
- Wedderburn, J. H. M. (1915). On matrices whose coefficients are functions of a single variable. *Transactions of the American Mathematical Society*, 16(3), 328–332.
- Wilson, E. B. (1990). *An introduction to scientific research*. Courier Corporation.
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.
- Wu, G., Wen, X., Wang, L., Pedrycz, W., & Suganthan, P. N. (2021). A Voting-Mechanism-Based Ensemble Framework for Constraint Handling Techniques. *IEEE Transactions on Evolutionary Computation*, 26(4), 646–660.

- Yang, Y., Zheng, K., Wu, C., Niu, X., & Yang, Y. (2019). Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. *Applied Sciences*, 9(2), 238.
- Yedjour, D., & Benyettou, A. (2018). Symbolic interpretation of artificial neural networks based on multiobjective genetic algorithms and association rules mining. *Applied Soft Computing*, 72, 177–188.
- Yusof, K. M., Isaak, S., Rashid, N. C. A., Ngajikin, N. H., & Bahru, U. J. (2016). NPK Detection Spectroscopy on Non-Agriculture Soil. *Jurnal Teknologi*, 78(11), 227–231.
- Zhang, L., Tan, J., Han, D., & Zhu, H. (2017). From machine learning to deep learning: Progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, 22(11), 1680–1685.
- Zheng, H., & Gu, Y. (2021). Encnn-upmws: Waste classification by a CNN ensemble using the UPM weighting strategy. *Electronics*, 10(4), 427.
- Zhou, Z. H. (2009). Ensemble learning, Encyclopedia of Biometrics. *Doi*, 10, 978–0.
- Zouggar, S. T., & Adla, A. (2018). A New Function for Ensemble Pruning. In F. Dargam, P. Delias, I. Linden, & B. Mareschal (Eds.), *Decision Support Systems VIII: Sustainable Data-Driven and Evidence-Based Decision Support*, 313, 181–190. Springer International Publishing. https://doi.org/10.1007/978-3-319-90315-6_15.

APPENDICES

Appendix 1: Research Validation Data Provision Letter

LIVE SUPPOST SYSTEM (T) LTD

Fax: +255 27 254499

Land Line: +255 27 254499

TFA HQ Plot No 76

Shule Road



Soil Care Department,

P.O. Box 1684,

ARUSHA.

25th April, 2023

To whom it may concern

Dear Sir/Madam

RE: PROVISION OF RESEARCH DATA

Mr. Augustine J Malamsha a student of Nelson Mandela African Institution of Science and Technology, Arusha, Tanzania, for research purposes, has been granted access to 48 soil samples from randomly selected from Njombe district. The samples were collected by using soil augers and packed in bags, before testing through the dry chemistry method. This data is therefore provided for validity checking purpose for the research study titled "*Weighted Voting Heterogeneous Hybrid Classifiers for High Performance Soil Fertility Status Prediction*". It is hoped that the data will be useful to the study.



(Ann Murray)

For : Soil testing Manager

Copy to:

Mr. Augustine J Malamsha

Student NM-AIST

Appendix 2: Python code for Fertility Index Derivation

```
Created on Sun Oct 18 22:49:06 2020
@author: Augustine J Malamsha
"""
#encoding="ISO-8859-1"
#encoding="cp1252"

import base_learners_evaluation
import streamlit as st
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from kneed import KneeLocator
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import seaborn as sns
from scipy.spatial.distance import cdist
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import statsmodels.api as sm
from statsmodels.formula.api import ols

def sfi_derivation():
    unlabeled_pedoagro_data = pd.read_csv('./_modelling_data/2data_nonredundant/agricultural_data.csv')
    #dropping some columns, getting lid of location aware
    uploaded_file = st.file_uploader("Select the past sanity check agricultural soils raw chemical data to upload")
    if uploaded_file is not None:
        dataframe = pd.read_csv(uploaded_file)
        #st.write(dataframe)
        st.dataframe(dataframe, 750, 150)
        unlabeled_pedoagro_data = dataframe
        unlabeled_pedoagro_data = unlabeled_pedoagro_data.drop(columns = ['SSN', 'Lat', 'Long', 'Ward'])
        unlabeled_pedoagro_Kmeandata = np.array(unlabeled_pedoagro_data['Maize_Yields']).reshape(unlabeled_pedoagro_data.shape[0],)

        #Elbow
        distortions = []
        inertias = []
        mapping1 = {}
        mapping2 = {}
        K = range(1,10)

    def cluster_locator(X):
        kmeans_kwargs = {
            "init": "random",
            "n_init": 10,
            "max_iter": 300,
            "random_state": 42,
        }

        sse = []
        for k in range(1, 11):
            kmeans = KMeans(n_clusters=k, **kmeans_kwargs)
            kmeans.fit(X)
            sse.append(kmeans.inertia_)

        kl = KneeLocator(range(1, 11), sse, curve="convex", direction="decreasing")

        return kl.elbow

    k=cluster_locator(unlabeled_pedoagro_Kmeandata)
    st.write('K value of the Knee-detection method=',k)
    kmeans = KMeans(n_clusters = k, max_iter=600, algorithm = 'auto')
    kmeans.fit(unlabeled_pedoagro_Kmeandata)
    Y = kmeans.labels_
    centroid = kmeans.cluster_centers_
    #Print the labels and centroids
    #st.write(Y)
    #st.write(centroid)

    #Append the labales column into the unlabelled data frame
    #And save the dataframe as a csv file
    unlabeled_pedoagro_data['yieldLbl'] = Y
    unlabeled_pedoagro_data = unlabeled_pedoagro_data.drop(columns = ['Maize_Yields'])

    unlabeled_pedoagro_data_transformed = unlabeled_pedoagro_data.drop(columns = ['yieldLbl'])
    unlabeled_pedoagro_data_transformed = pd.DataFrame(preprocessing.Normalizer().fit_transform(unlabeled_pedoagro_data_transformed))
    unlabeled_pedoagro_data_transformed['yieldLbl'] = unlabeled_pedoagro_data['yieldLbl']
    unlabeled_pedoagro_data_transformed.columns = unlabeled_pedoagro_data.columns
    unlabeled_pedoagro_data.to_csv("./_modelling_data/2data_nonredundant/Labelled_soil_yield_data.csv",index=False)
    unlabeled_pedoagro_data_transformed.to_csv("./_modelling_data/2data_nonredundant/Labelled_soil_yield_data_transformed.csv",index=False)
```


Appendix 3: Python code for Base Models Evaluation

```
for name, clf in tqdm(zip(names, classifiers), desc=" ...Evaluation: <<Testing>>"):
    model = clf.fit(X_train, y_train)
    model_sm = clf.fit(X_train_sm, y_train_sm)
    if name == "KNN":
        pickle_out = open("./models_depl/kn_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "SVM":
        pickle_out = open("./models_depl/sv_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "DT":
        pickle_out = open("./models_depl/dt_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "RF":
        pickle_out = open("./models_depl/rf_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "AdaB":
        pickle_out = open("./models_depl/ab_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "NB":
        pickle_out = open("./models_depl/nb_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    elif name == "GB":
        pickle_out = open("./models_depl/gb_learner.pkl", "wb")
        pickle.dump(model, pickle_out)
        pickle_out.close()
    predictions = model.predict(X_test)
    score = accuracy_score(y_test, predictions)
    learner.append(name)
    base_score.append(score)
    high_prec.append(precision_score(y_test, predictions, average=None)[2])
    mid_prec.append(precision_score(y_test, predictions, average=None)[1])
    low_prec.append(precision_score(y_test, predictions, average=None)[0])
    high_recall.append(recall_score(y_test, predictions, average=None)[2])
    mid_recall.append(recall_score(y_test, predictions, average=None)[1])
    low_recall.append(recall_score(y_test, predictions, average=None)[0])
    high_f1.append(f1_score(y_test, predictions, average=None)[2])
    mid_f1.append(f1_score(y_test, predictions, average=None)[1])
    low_f1.append(f1_score(y_test, predictions, average=None)[0])
    wgted_prec.append(precision_score(y_test, predictions, average='weighted'))
    wgted_recall.append(recall_score(y_test, predictions, average='weighted'))
    wgted_f1_score.append(f1_score(y_test, predictions, average='weighted'))
```

Appendix 4: Python code for 1EXP (-) Z+ Initial-Term Based Arithmetic Sequences formulation and weights coefficients generation function algorithm

```
#SUBMOD01:-----
def generate_dim_ratios(n):
    switcher = {
        1: 1e-01,
        2: 1e-02,
        3: (1e-02)/2,
        4: (1e-02)/4,
        5: (1e-02)/8,
        6: (1e-02)/16,
        7: (1e-02)/32,
        8: (1e-02)/64,
        9: (1e-02)/128,
        10: (1e-02)/256,
        11: (1e-02)/512,
        12: (1e-02)/1024,
    }

    return switcher.get(n, "Invalid space")

#SUBMOD02:-----
def generate_weights_vector(ranks_prec_degree):
    #Generating initial weights distributions
    default_ratio = generate_dim_ratios(ranks_prec_degree)
    weights_vector = []
    weights_vector.append(0.0)
    for i in stqdm(range(0, int(1/default_ratio) - 1), desc="... *ai.in ---> Ensemble Optimization Weights values Grid built-in Gen"):
        rational_wts = default_ratio + (default_ratio * i)
        weights_vector.append(round(rational_wts, 12))
    return default_ratio, weights_vector

#SUBMOD03:-----
cw_domain_type = []
cw_domain_size = []
def create_MatrixA_clrsweights(weights_vector, ci, ens_no): #The benchmark for automated EnOW for use in EnOWasBenchmark_GridSearchEnOpt
    'maxrnk1_Perm_CLRSWGHTS_mxn'
    clrs_weights_matrix = []
    #RETURNS ALL POSSIBLE COMBINATIONS AS AN ITERTOOL TO APPEND INTO A LIST AS VALUES
    base_weight_vector = permutations(weights_vector, len(ci)) #Permute weights vector into number of columns = to the number of base learners
    # for i in stqdm(list(base_weight_vector), desc="... FPP asymptotic function algorithm formulating base learners weights domains"):
    for i in stqdm(list(base_weight_vector), desc="... base Learners weights values domains/grid formulation"):
        clrs_weights_matrix.append(i)
    if ens_no == 1:
        clrs_weights_matrix = pd.DataFrame(clrs_weights_matrix, columns=['gb', 'rf', 'sv', 'kn'])
        # Taking only the rows whose ranking totals to 1
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['rf'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['kn'] != 0)]
    elif ens_no == 7:
        clrs_weights_matrix = pd.DataFrame(clrs_weights_matrix, columns=['sv', 'kn'])
        MAT_A_CLRSWGHTS_mxn = clrs_weights_matrix[(clrs_weights_matrix['sv'] + clrs_weights_matrix['kn']) == 1]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['sv'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['kn'] != 0)]
    elif ens_no == 8: #ens_no 8 ----> gb rf sv
        clrs_weights_matrix = pd.DataFrame(clrs_weights_matrix, columns=['gb', 'rf', 'sv'])
        MAT_A_CLRSWGHTS_mxn = clrs_weights_matrix[(clrs_weights_matrix['gb'] + clrs_weights_matrix['rf'] + clrs_weights_matrix['sv']) == 1]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['gb'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['rf'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['sv'] != 0)]
        clrs_weights_matrix.to_csv("../modelling_results/auto/3bm-initial.csv", index=False)
        MAT_A_CLRSWGHTS_mxn.to_csv("../modelling_results/auto/3bm-constrained.csv", index=False)
        cw_domain_type.append("3bm-initial")
        cw_domain_size.append(clrs_weights_matrix.shape[0])
        cw_domain_size.append("3bm-constrained")
    elif ens_no == 9: #ens_no 8 ----> gb rf kn
        clrs_weights_matrix = pd.DataFrame(clrs_weights_matrix, columns=['gb', 'rf', 'kn'])
        MAT_A_CLRSWGHTS_mxn = clrs_weights_matrix[(clrs_weights_matrix['gb'] + clrs_weights_matrix['rf'] + clrs_weights_matrix['kn']) == 1]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['gb'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['rf'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['kn'] != 0)]
    elif ens_no == 10: #ens_no 10 ----> rf sv kn
        clrs_weights_matrix = pd.DataFrame(clrs_weights_matrix, columns=['rf', 'sv', 'kn'])
        MAT_A_CLRSWGHTS_mxn = clrs_weights_matrix[(clrs_weights_matrix['rf'] + clrs_weights_matrix['sv'] + clrs_weights_matrix['kn']) == 1]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['rf'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['sv'] != 0)]
        MAT_A_CLRSWGHTS_mxn = MAT_A_CLRSWGHTS_mxn[(MAT_A_CLRSWGHTS_mxn['kn'] != 0)]
    #End of making a a dictionaries with a list of base learners and weight paramter values with key as base learners and weight values
    return MAT_A_CLRSWGHTS_mxn # returns constrained weights MAT_A_CLRSWGHTS_mxn as the EnOW
```

Appendix 5: Python code for the complete WVE brute exhaustive Optimization module

```
def enweight_benchmark_brutearch(_DATA,clrs_names,weights_vector,ens_no):
    ci = gen_classifiers_initials(clrs_names)
    MAT_A_CLRSWGHTS_mxn = create_MatrixA_clrsweights(weights_vector,ci,ens_no)
    #Matrix B -- transposed permutations of probability predictions
    probpred_list = gen_clrs_proba_predictions(_DATA)
    if ens_no == 1:
        gb = np.asarray(probpred_list[0])
        rf = np.asarray(probpred_list[1])
        sv = np.asarray(probpred_list[2])
        kn = np.asarray(probpred_list[3])
        linker_dict = {'gb':gb,'rf':rf,'sv':sv,'kn':kn}
        ci_of_probpred = ['gb','rf','sv','kn']
    elif ens_no == 2:
        gb = np.asarray(probpred_list[0])
        rf = np.asarray(probpred_list[1])
        linker_dict = {'gb':gb,'rf':rf}
        ci_of_probpred = ['gb','rf']
    elif ens_no == 3: #ens_no 2 ----> gb rf
        gb = np.asarray(probpred_list[0])
        sv = np.asarray(probpred_list[1])
        linker_dict = {'gb':gb,'sv':sv}
        ci_of_probpred = ['gb','sv']
    elif ens_no == 4:
        gb = np.asarray(probpred_list[0])
        kn = np.asarray(probpred_list[1])
        linker_dict = {'gb':gb,'kn':kn}
        ci_of_probpred = ['gb','kn']
    elif ens_no == 5: #ens_no 5 ----> rf sv
        rf = np.asarray(probpred_list[0])
        sv = np.asarray(probpred_list[1])
        linker_dict = {'rf':rf,'sv':sv}
        ci_of_probpred = ['rf','sv']
    elif ens_no == 6:
        rf = np.asarray(probpred_list[0])
        kn = np.asarray(probpred_list[1])
        linker_dict = {'rf':rf,'kn':kn}
        ci_of_probpred = ['rf','kn']
    elif ens_no == 7:
        ci_of_probpred = ['gb','rf','kn']
    elif ens_no == 10: #ens_no 10 ----> rf sv kn
        rf = np.asarray(probpred_list[0])
        sv = np.asarray(probpred_list[1])
        kn = np.asarray(probpred_list[2])
        linker_dict = {'rf':rf,'sv':sv,'kn':kn}
        ci_of_probpred = ['rf','sv','kn']

    a = list(permutations(ci_of_probpred))
    b = list(permutations(ci_of_probpred))
    for i in stqdm(range(len(a)), desc = " Creating Vectorization arrays: Linking candidate Ensemble combination to permuted base models"):
        a[i] = list(a[i])
        for j in range(len(a[i])):
            a[i][j] = linker_dict[a[i][j]]
        sleep(0.3)
    a_matrix = a
    MAT_A_CLRSWGHTS_mxn = np.asarray(MAT_A_CLRSWGHTS_mxn)
    MAT_A_1stexp = np.expand_dims(MAT_A_CLRSWGHTS_mxn,axis = 0)
    MAT_A_2ndexp = np.expand_dims(MAT_A_1stexp,axis = 0)
    MAT_A_2ndexp_rs = np.reshape(MAT_A_2ndexp,(MAT_A_CLRSWGHTS_mxn.shape[0],MAT_A_CLRSWGHTS_mxn.shape[1],1,1))
    max_score, i_val, j_val = 0,0,0
    for i in stqdm(range(MAT_A_2ndexp_rs.shape[0]),desc ="ENSEMBLE OPTIMIZATION(GridSearch):____->>i) SIMD_Vectorization [<< SI: PROBABILITIES]):
        out1 = MAT_A_2ndexp_rs[i] * a_matrix
        out2 = out1.sum(axis = 1)
        for j in range(out2.shape[0]):
            score = accuracy_score(np.argmax(out2[j],axis = 1),_DATA[3]) #SCORING THE ENSEMBLE
            if score > max_score:
                max_score = score
                i_val = i
                j_val = j
                clrs_comb = list(b[j_val])
                wts_comb = MAT_A_CLRSWGHTS_mxn[i_val]
    return clrs_comb,wts_comb,max_score #Returing the softvoted ensemble INIT. weights and evaluation score
```

Appendix 6: Python code for ROC Analysis

```
ANALYSE ENSEMBLE ROC
#Voting the Ensemble
prediction_en_roc = (wt_gb * prediction_engb_roc) + (wt_rf * prediction_enrf_roc)\
    + (wt_sv * prediction_ensv_roc) + (wt_kn * prediction_enkn_roc)

# Compute ROC curve and ROC area for each class
fpr_gbrfknsv = dict()
tpr_gbrfknsv = dict()
roc_auc_gbrfknsv = dict()

for i in stqdm(range(n_classes), desc="Evaluating GB,RF,SV, and KNN ensemble's discrimination ability of the fertility
    fpr_gbrfknsv[i], tpr_gbrfknsv[i], _ = roc_curve(y_test_roc[:, i], prediction_en_roc[:, i])
    roc_auc_gbrfknsv[i] = auc(fpr_gbrfknsv[i], tpr_gbrfknsv[i])

# Compute micro-average ROC curve and ROC area
fpr_gbrfknsv["micro"], tpr_gbrfknsv["micro"], _ = roc_curve(y_test_roc.ravel(), prediction_en_roc.ravel())
roc_auc_gbrfknsv["micro"] = auc(fpr_gbrfknsv["micro"], tpr_gbrfknsv["micro"])

# Compute macro-average ROC curve and ROC area
# First aggregate all false positive rates

all_fpr = np.unique(np.concatenate([fpr_gbrfknsv[i] for i in range(n_classes)]))

# Then interpolate all ROC curves at this points
mean_tpr = np.zeros_like(all_fpr)
for i in range(n_classes):
    mean_tpr += interp(all_fpr, fpr_gbrfknsv[i], tpr_gbrfknsv[i])

# Finally average it and compute AUC
mean_tpr /= n_classes
fpr_gbrfknsv["macro"] = all_fpr
tpr_gbrfknsv["macro"] = mean_tpr
roc_auc_gbrfknsv["macro"] = auc(fpr_gbrfknsv["macro"], tpr_gbrfknsv["macro"])
```

Appendix 7: Python code for the WVE optimization data loading and Main Module

```
#MAIN FUNCTION:-----
def main():
    _DATA,X_roc,y_roc= read_and_split_input_datafile()
    # Binarize the output
    y_roc = label_binarize(y_roc, classes=[0, 1, 2])
    n_classes = y_roc.shape[1]
    X_train_roc, X_test_roc, y_train_roc, y_test_roc = train_test_split(X_roc, y_roc, test_size=.3,
                                                                           random_state=0)

    def model_f(value):
        if value == 0:
            model = GradientBoostingClassifier(max_depth=15, loss="deviance",learning_rate = 0.1,max_features = "sqrt", criterion = "fr
        if value == 1:
            model = RandomForestClassifier(max_depth=19, max_leaf_nodes = 1300,random_state =123)
        if value == 2:
            model = SVC(kernel = 'rbf', probability=True, gamma=3350, C=2,random_state =123)
        if value == 3:
            model = KNeighborsClassifier(leaf_size = 25, n_neighbors = 1,p=1,algorithm='auto')
        return model

    granularity = st.number_input("Select weights values granularity degree ", 1, 12, 1, 1)
    if granularity > 3:
        st.error("Quantum Computational settings required: The selected degree is under versioning, Caution: Extreme Resources Intensive")
    else:
        if granularity == 1:
            st.write("Granularity 1:- Less computational time with a minimal of Core Dual 7 processor")
        if granularity == 2:
            st.write("Granularity 2:- Much computational time is required and a minimal of Core Dual 7 processor")
        if granularity == 3:
            st.warning(" Caution:- Computational Resources Intensive e.g not recommended for non-HPC machine")
        if st.button("RUN"):
            default_ratio, weights_vector = generate_weights_vector(granularity)
```

RESEARCH OUTPUTS

International Journals

Malamsha, A. J., Dida, M. A., & Moebs, S. (2023). Brute Exhaustive Optimization of Intelligent Small Weighted Voting Ensembles in $1EXP(-)Z^+$ Initial-Term based Arithmetic Sequence's Multi Precision Search Spaces. *Journal of Mathematics and Informatics*. 25, 29-46.

International Conferences

Malamsha, A. J., Dida, M. A., & Moebs, S. (2023). 2-Stage Hybrid Based Heterogeneous Ensemble Committee Machine for Improving Soil Fertility Status Prediction Performance. *International Conference for Technological Advancement in Embedded and Mobile Systems*. ICTA-EMOS.

Poster Presentations

Malamsha, A. J., Dida, M. A., & Moebs, S. (2023). 2-Stage Hybrid Based Heterogeneous Ensemble Committee Machine for Improving Soil Fertility Status Prediction Performance. *International Conference for Technological Advancement in Embedded and Mobile Systems*. ICTA-EMOS.

Book Section

Malamsha, A. J., Dida, M. A., & Moebs, S. (2023). 2-Stage Hybrid Based Heterogeneous Ensemble Committee Machine for Improving Soil Fertility Status Prediction Performance. *Artificial Intelligence Tools and Applications in Embedded and Mobile Systems*. Springer Nature.

Pre-Print

Malamsha, A. J., Dida, M. A., Moebs, S. (2023). A Survey of Machine Learning Modelling for Agricultural Soil Properties Analysis and Fertility Status Predictions. *Preprints 2023*, 2023081395. <https://doi.org/10.20944/preprints202308.1395.v1>