Augmented Intelligence Approach to Educational Data Mining: Student Drop Prediction

by

Keegan Freeman

A thesis submitted to the Cullen College of Engineering,

in partial fulfillment of the requirements for the degree of

Bachelor of Science

in Computer Engineering

Chair of Committee: Dr. Nouhad Rizk

Co-Chair of Comittee: Dr. Shishir Shah

Committee Member: Dr. Xin Fu

University of Houston

May 2021

Acknowledgements

First and foremost, I would like to acknowledge my thesis director, Dr. Nouhad Rizk, for her overwhelming support in crafting my thesis as well as her guidance in data science. Dr. Rizk was a large influence in my pursuit of graduate studies in data science, and as a result I will be attending the University of Washington in Fall of 2021. Dr Rizk was incredible at helping me find research materials that wound up being the basis of this research. I am incredibly grateful to have learned data science from both Dr. Rizk's course and through this honors thesis that allowed me to get my hands wet in the field.

Secondly, I would like to thank Dr. Heidel for her aid in both streamlining my thesis methodology as well as her support in getting my survey distributed to the LAUNCH tutoring center's students.

Next, I would like to thank my rescue cat, Blue, for sitting on my keyboard and giving me a much-needed break often throughout this last year. Blue has witnessed more of my thesis writing than anyone else, and despite not knowing much about it, would recognize its impact on me when stressful, providing encouraging headbutts.

Next, I would like to thank Dr. Nguyen who taught my first course in introduction to machine learning and gave me an introduction into many of the tactics I implemented in this research.

Finally, I would like to express my gratitude to all of my committee members for their support through the years in my machine learning progress. Dr. Shah, sponsored my SURF and PURS research and allowed me to get my first experience with machine learning. Dr. Xin Fu agreed to be a committee member during a hectic semester. Finally, I would like to thank Dr. Leonard Trombetta for his overwhelming academic assistance in being understanding of my electric academic interests, primarily in computer science. Dr. Leonard Trombetta taught many of my ECE courses, but beyond that has been an invaluable resource to my success here at the University of Houston.

Last but not least, I would like to thank the Honors College for giving me the opportunity to participate in this research as well as help develop me into a critically thinking person.

Abstract

Educational Data Mining (EDM) and Augmented Intelligence (AUI) are two upcoming fields in the machine learning research industry. EDM refers to the use of machine learning elements in an educational format. Typically, this is in the form of utilizing educational data to better understand the learning process. Augmented Intelligence, on the other hand, is a niche of machine learning that refers to people taking a much larger role than typical in artificial intelligence projects. For example, a professional in a given field may provide better insight as to what metrics should be weighed more when considering a given prediction.

In this thesis, I review the feasibility of using Augmented Intelligence in the genre of Educational Data Mining to predict the likelihood of a student dropping a course based on demographic, study habit, and student perception information recorded through a survey. Additionally, I will be testing three optimization algorithms to see which is most beneficial in the application of this research. The goal of this research is to ultimately provide instructors with a machine learning model capable of highlighting at risk students such that the instructor can provide intervention techniques in a more timely fashion.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 -- Problem Background

Educational data mining (EDM) is a field of increasing popularity among teachers for its ability to make predictions regarding a student's performance. Typically, EDM is used to help teachers identify struggling students in time for teacher recommended practices to reverse the performance decline of the highlighted students. Similarly, augmented intelligence is a topic of increasing relevance as the ability to directly intervene in machine learning modeling is seen as beneficial, especially in regard to smaller datasets. The addition of augmented intelligence to educational data mining could produce more accurate results than typical EDM approaches, allowing for the modeling of student performance to be a more reliable resource for teachers seeking to improve their class.

1.2 -- Problem Statement

The fundamental aspect of this research is to accurately predict which students are likely to drop a course based on a variety of factors including educational, family, job, and miscellaneous influences. This prediction model will be novel in that it will incorporate augmented intelligence to combat a limited dataset representing the students of a class. Furthermore, the predictions will be modeled through easy-to-read labels such that teachers with little experience in data science may benefit from my research. I aim for my research to provide insight as to successful modeling of augmented intelligence as well as aid teachers in recommending beneficial support programs for those deemed to drop.

1.3 -- Research Question

Question that is being used to guide this study:

1. Can the combination of augmented intelligence and educational data mining provide reasonably accurate predictions of student drop likelihood?

Delimitations

1. The study is limited to the students that are enrolled in the Bachelor of Science, Computer Science or Engineering programs as well as those enrolled in the LAUNCH tutoring center at the University of Houston-Main campus.
2. This study will not include post baccalaureate students.
3. This study is limited to students enrolled in COSC 1430, COSC 2430, ENGI 1331, engineering majors, and students enrolled in the LAUNCH tutoring center at the University of Houston.
4. This study does not offer differentiation between classes with grade curves and classes without grade curves.

Assumptions

1. The students participating in the study adhere to the University of Houston academic honesty policy.
2. The data collected from students is accurate.

Definitions

1. Students that "drop" are those considered to have taken a "W" prior to the University of Houston's official drop date.

1.4 -- Research Design

Essential Data Analysis Questions:

1. Variables that will be used in this research fall under the following categories: demographic, educational, and individual perception.
2. The overall objective of this research is to accurately predict the likelihood of students dropping a course based on previously mentioned variables. This research will include

optimizer experimentation to find the best suited optimizer for prediction using the classroom sized dataset.

Outline of Optimizers that will be used in the Analysis of Data:

1. Several optimizers will be taken into consideration for the purpose of this research. Those being tested are Adagrad, Lion Optimization Algorithm, Grey-Wolf Optimizer, and the Lion-Wolf Optimizer.

Outline of Machine Learning Models that will be used in the Analysis of Data:

1. Several machine learning models will also be taken into consideration for the purpose of this research. The differentiation between an optimizer and machine learning model is made later in this paper. The models being tested are Support Vector Machine, Gaussian Naïve Bayes, and a neural network.

1.5 -- Literature Review

1.5.1 -- Educational Data Mining

Educational data mining (EDM) is an area of data science related to analyzing educational data in order to provide a better understanding of students and how they learn. A hierarchy in educational data is a commonly used tool for educational data mining. Furthermore, EDM often integrates both machine learning and data mining to make predictions about students. Differing from most data mining, the datasets used by EDM are often small, about the size of a classroom, when compared to typical data mining uses. This trademark of EDM provides an obstacle for correctly implementing machine learning prediction as most techniques require larger datasets. Four commonly used applications of EDM include improving student modeling of performance, improving models of knowledge structure, analyzing effectiveness of student support initiatives, and scientific discovery about the learning process [1].

Educational data mining (EDM) has recently gained popularity for its benefits including offering

suggestions to academic planners and enhance teacher's decision-making process for underperforming
students. Student performance prediction is a tool commonly used in EDM. These models are based on
a plethora of factors including societal, school, college, individual, and family. Recently neural network
models have made their way to EDM through performance prediction bolstered by the Statistical Package for Social Sciences [2].

While most EDM utilizes a type of unsupervised learning through clustering, this research will novel in that it uses supervised learning as a means to predict student performance rather than be used to find associations between parameters [3].

1.5.2 -- Augmented Intelligence

As A.I and machine learning become more commonplace in professional environments, an incentive to restructure the human-oriented business environment is rising. This new structure will need businesses to determine which tasks should be left at the discretion of machines, humans, or a mixture of the two – otherwise known as augmented intelligence. The concept of augmented intelligence revolves around A.I. bolstering human capabilities such that the strengths of A.I and human abilities are both utilized as to produce maximum efficiency. One of the many human strengths, innovation, is already implementing A.I. in new and creative ways in different business fields such as medical, healthcare, legal, and finance. One of the benefits of augmented intelligence is utilizing big data and A.I. to give humans a better understanding of their customer base. As more jobs are being automated, the future of work will be centered around human and machine cooperation. The next step in our capitalistic and technologically advancing society is to further encourage innovation through augmented intelligence. Employees must become better educated to work in a society pushing towards augmented intelligence. [4].

1.5.3 -- Neural Network Overview

Neural networks (NN) are structurally based on the anatomy of the human brain. Neurodes are organized into a series of interacting layers. Three types of layers exist in a neural network: input, output, and hidden. Input and output layers are self-explanatory as they receive data input and produce an output respectively. Hidden layers are what contain different weights to represent an equation using the given input to produce the desired output [2]. In the scope of this research, each optimizer can be thought of as the series of interconnected hidden layers. Together, they will produce an output used to predict a result based on the input to the pipeline. The typical architecture of a neural network can be visualized in the following figure.

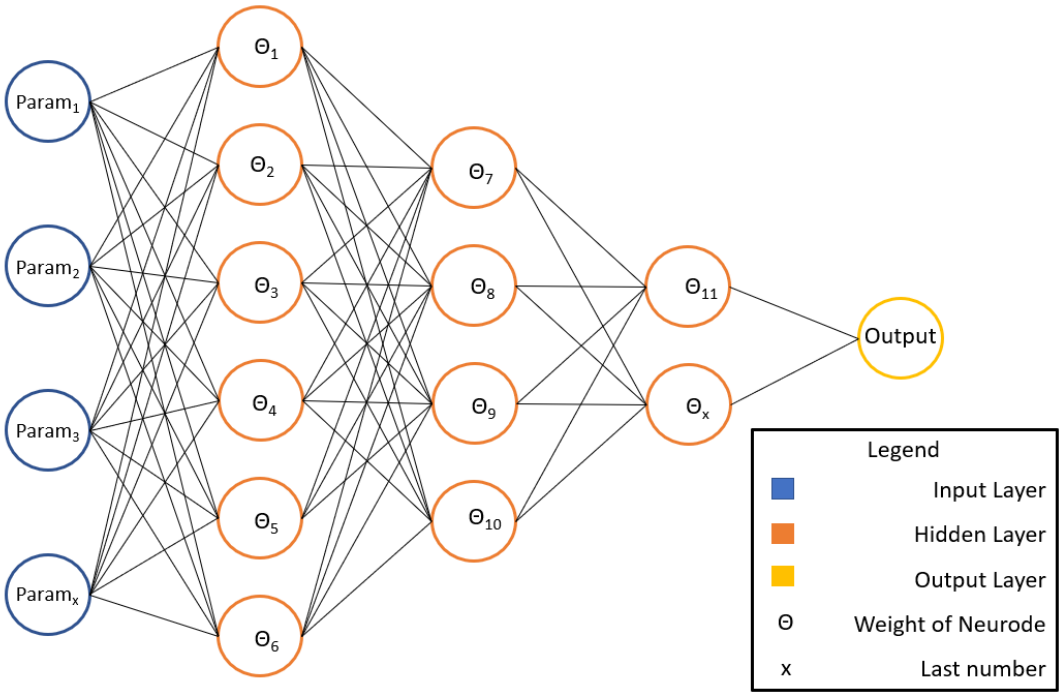

Figure 1: Neural Network Visualization

1.5.4 -- Gradient Descent Overview

Gradient descent utilizes back propagation -- the process of propagating input through a neural network from the first layer to the last, calculating the loss using a predetermined function, and finally updating the weights of each neurode by backpropagating the loss from the last layer to the first [5]. The goal of gradient descent is to minimize the cost function, also referred to as loss,

by updating each parameter's weight [5]. The weight of a parameter can be thought of as the slope or multiplier of a variable in an equation. The size of the steps a weight can change is referred to as the learning rate, commonly denoted as η [6]. By the nature of gradient descent, a parameter is updated only once for a dataset during backpropagation, and therefore converges to a solution at a slow rate [5]. Many variants of gradient descent have been developed to combat slow convergence and other issues associated with the base algorithm, one of which is Adagrad.

1.5.5 -- Metaheuristic Algorithm Overview

Metaheuristic algorithms are defined as those inspired by natural phenomena such as nature, physics, mathematics, animal sociology, and politics [7]. Metaheuristic algorithms are designed to optimize problems with discrete or continuous variables, making them applicable to a wide variety of problem types [7]. Characteristics of metaheuristic algorithms are as follows:

- Lack of complex mathematical expressions [7]
- Commonly model social structure of animals [7]
- Robust in application [7]
- Well-suited for problems with expensive or unknown derivative information [8]

The base structure for a metaheuristic algorithm is as follows:

1) Creation of base vectors [7]
2) Evaluation of base vectors [7]
3) Creation of new set of vectors [7]
4) Evaluation of new vectors [7]
5) Vector Comparison [7]
6) Redefinition of Step-Size [7]
7) Termination Criteria Analysis [7]

1.5.6 -- Adagrad Optimizer

Adagrad, also known as the adaptive gradient algorithm, is a variation of the commonly used gradient descent optimization algorithm used in neural networks. Practically, these both

function as the "learner" for a constructed neural network as they strive to produce weights for an equation that results in the least "loss". To better understand Adagrad, gradient descent should first be overviewed.

Adagrad Overview

Adagrad is a variant of the gradient descent optimization algorithm most notable for its unique learning rates for each parameter [6]. Because of its adaptive learning rate, Adagrad is an appropriate approach to working with sparse data [6]. Another key feature of Adagrad is it removes the need to manually alter the learning rate as is the case for basic gradient descent [6]. The largest disadvantage that lies within the Adagrad optimizer is that due to the nature of its adaptive learning rate equation, there comes a point when the optimizer will essentially stop learning given too much data. This disadvantage is due to the denominator of the equation being based on each and every datapoint in a cumulative fashion such that it will eventually converge to zero [6].

1.5.7 -- Naive Bayes

Naïve Bayes is a supervised learning algorithm based on the Bayes' Theorm shown below.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Figure 2: Bayes Theorem

Essentially, Bayes theorem calculates the probability of the class A given an object B [9].This process translates to a classification algorithm when implemented in a machine learning pipeline. Naïve Bayes has a linear training time based on the quantity of dimensions and training points because of its lack of searching [9]. Furthermore, this machine learning pipeline is characteristic of low variance but high bias, again, due to its lack of data searching [9]. Being based in a statistical probability equation, this machine learning process is rather robust to noise, as it accounts for irregularities [9].

1.5.8 -- SVM

A support vector machine (SVM) is a type of classification algorithm used in machine learning. This process is supervised, meaning it learns from examples and their associated labels. The novel aspect behind SVM is that it aims to separate classes using a hyperplane with a number of dimensions equal to that of the objects fed into the pipeline [10]. Through a given number of iterations, SVM will calculate a variety of hyperplanes and determine their success based on the number of correctly associated classes on either side with as little overlap as possible [10]. Variations of SVM include non-linear hyperplanes, dubbed soft margins, and kernels that add dimensions to allow for greater separability at the cost of a greater number of solutions [10]. While a competitive tool for classification, SVM will decrease in accuracy with a greater number of dimensions for each data point.

1.5.9 -- Lion Optimization Algorithm

The Lion Optimization Algorithm (LOA) is a metaheuristic algorithm meaning it is well-suited for sparse and incomplete datasets. Furthermore, this algorithm is structured such to model the social structure of lions in the wild. The key elements taken from the lion social behavior are as follows:

- A solution representative of a territorial lion should be stronger than, synonymous to more accurate than, a random solution representative of a nomadic lion [11]

- Weak solutions representative of weak lions or cubs should adhere to Darwinism and be removed from the population [11]
- Solutions derived from success are more accurate and stronger than those derived from failure [11]

Here, each solution represents a point in some defined bounds. This point can then be converted to a value using an equation such as $x_i = a_i + b_i$ where *i* would range from 1 to the total number of parameters.

The algorithm pipeline for the Lion Optimization Algorithm is as follows:

I. Initialization

- A set of parameter weights are defined as a lion. The fitness value of a lion is akin to the cost function in neural networks. With this in mind, lions are generated randomly to the determined population and each assigned a gender, pride, and role. Predefined ratios determine the number of lions for each gender, pride, and role. Lastly, a territory is formed for each pride representing an area within the range of the defined bounds for the parameters [12].

II. Hunting

- Females are designated hunters and therefore target prey. Hunters are divided into three subgroups randomly and use the hunting strategy used by actual lions by encircling their prey. During movement from hunting, akin to solution finding, if a hunter improves its fitness then they prey escapes to a new location. Benefits of this hunting strategy include allowing different directions to be used while moving towards prey as well as solutions to escape from local optima [12].

III. Moving to a Safe Place

- The pride's territory will vary based on hunting from the females. Success rate is calculated using the previous and current iterations locations. High success indicates convergence towards a point far from the optimal solution.

Conversely, low success indicates territory movement around the optimal solution but without substantial improvement. Location diversity will adapt to the success rate iteratively such to find optimal solutions [12].

IV. Roaming

- Roaming occurs in male lions part of a pride. The bounds of this roaming lie within the pride's territory. The goal of males roaming is to find better positions within the territory. This serves as a local search for optimal positions within the territory. Similarly, nomad lions roam randomly across the entire area of the determined bounds.

V. Mating

- For every pride, a percentage of the female lions mate with one or more resident males selected at random. For nomads, females will mate with only one other male lion selected at random across the total area. For every act of mating, 2 cubs are produced using a linear combination of the parents. Each of the two cubs is assigned opposite genders at random. Furthermore, mutations are simulated by a percentage of the cubs receiving a random number for a parameter weight instead of being based on the parents [12].

VI. Defense

- Once they reach maturity, lions will fight other males within their pride. Similarly, nomad male lions may also fight other males across all prides. Beaten males will become nomads and the victors will become resident males of the pride [12].

VII. Migration

- To increase diversity among tribes, a percentage of females of each tribe will be selected to become nomads. Furthermore, fit nomadic lioness' will migrate to prides, filling the place of residents lost [12].

VIII. Lion Population Equilibrium

- To maintain population equilibrium, nomad lions with low fitness will be removed from the total population to account for incoming cubs.

IX. Convergence

- The best result is calculated among the total population upon stopping criteria such as CPU time, maximum number of iterations, or number of iterations without improvement [12].

1.5.10 -- Grey-Wolf Optimization Algorithm

The Grey Wolf Optimization (GWO) Algorithm is yet another metaheuristic algorithm, making it suitable for sparse data. Similar to the Lion Optimization Algorithm, GWO is also inspired by nature, more specifically, Grey Wolves. Grey wolves have a social structure as follows that GWO mimics:

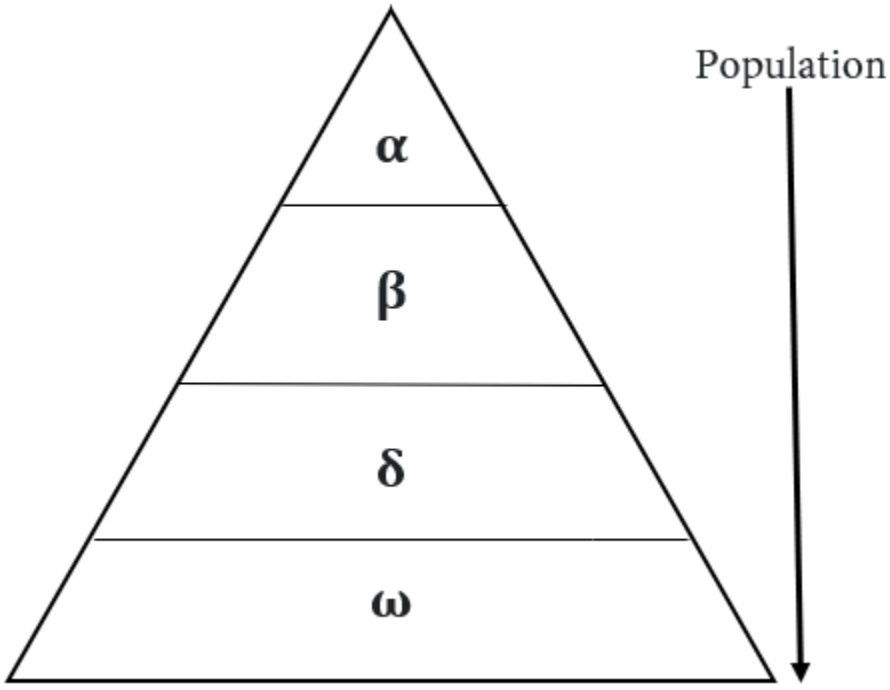

Figure 3: Grey Wolf Hierarchy

Alphas, the top of the hierarchy, consist of a single male and female. They are in charge of making decisions regarding hunting, location, and other large impact responsibilities. While the alpha male is not necessarily the strongest, it is the greatest at managing the pack and its decisions. The beta wolves,

the second in the hierarchy, aid the alphas in making pack decisions. They are also second in line following the decease of one of the alphas. Omega wolves, the bottom of the hierarchy, serve as an outlet for frustration among other wolves. This role aids in keeping peace among the pack and helps reduce infighting [8].

The key elements of the Grey Wolf Optimization Algorithm are highlighted below:

I. Social Hierarchy Instantiation

- The most fit solution is designated alpha. The second and third best solutions are deemed beta and delta respectively. Finally, all other solutions are considered omega [8].

II. Encircling Prey

- In the wild, grey wolves encircle their prey during hunting. A mathematical formula is used to simulate this throughout the parameter area similar to that of the Lion Optimization Algorithm [8].

III. Hunting

- While in the wild, Grey Wolves are capable of determining the location of their prey, this is not feasible to translate to an optimizer. Instead, the top three solutions are used to generate the location of prey. The encircling tactic can be implemented once pretty location is determined [8].

IV. Searching for Prey

- Grey wolves diverge to scour an area for prey. This concept is actualized using a randomized direction vector to simulate searching for prey for each wolf in the

pack. This is used to avoid local optima most likely found using the alpha, beta, and delta solutions [8].

Pseudocode of the Grey-Wolf Optimizer is shown in figure 3.

```
Initialize population of grey wolf pack
Calculate fitness of each solution to determine alpha, beta, and delta
while (t < interation max)
        for each search agent
                Update position of current search agent
        end
        Calculate fitness of all search agents
        Redetermine alpha, beta, and delta solutions
        t++
end
return alpha solution
```

Figure 4: Pseudo code of Grey Wolf Optimizer [8]

1.5.11 -- Lion-Wolf Hybrid Optimization Algorithm

As the title suggests, this algorithm is based on the metaheuristic Lion Optimization and Grey Wolf Optimization algorithms. This hybrid primarily leans towards the Grey Wolf Optimization algorithm but incorporates the Lion Optimization's position updates alongside that of the Grey Wolf's. This tweak of updating positioning results in an increase in convergence in addition to a better avoidance of local optima. As with the prior two optimizers, a solution is representative of a solution vector containing weights for each parameter [2]. The algorithm pipeline is as follows:

I. Parameter Setup
    - Create coefficient vectors representing position updates for each wolf/solution. Termination criteria is also initialized through a maximum number of iterations [2].

II. Population Setup
    - Create set number of solutions using a randomizer to initialize weights [2].

III. Fitness Calculation

- Using determined cost function, determine fitness of each wolf and assign the top three to alpha, beta, and delta, respectively [2].

IV. Lion Fusion

- Assign a fertility to each female based on tweaked Lion Optimization algorithm [2].

V. Lion-Wolf hybridization

- Using Lion Optimization hierarchy assignment, determine beta wolf to be second-most fit according to Lion algorithm [2].

VI. Position Updating

- Use the Grey-Wolf technique to update positions of solutions [2].

VII. Solution Updating

- Using the Grey-Wolf technique update each solution and reassign hierarchy [2].

VIII. Iterate

- Continue iterations until termination criteria is met [2].

1.5.12 -- Active Learning

Overview

Active learning is a machine learning technique primarily used to combat a lack of data in a machine learning model [13]. Active learning is a semi-supervised strategy that combats limited, incomplete, or unlabeled datasets. Typically this is achieved by focusing on the limited number of labeled, or user-queried, data in order to classify the remaining data-points, however other strategies such as data construction exist [14].

Characteristics of Active Learning

Common applications of active learning include data classification, such as text and image, as well as object detection such as pedestrian, network intrusion, and others [14]. The key concept behind active learning is allowing an algorithm to choose the data it deems worthy of "learning" from

[13]. Typical active learning frameworks contain a function to determine the informativeness of unlabeled data [15].

Querying

Three types of querying are primarily used for active learning, all of which utilize the informativeness of a data point. These query types are stream-based, membership, and pool-based [13]. Stream-based querying assumes low cost of labeling data, so it determines whether each data point is worth classifying or should be rejected based on its informativeness [15]. Membership querying is a process of allowing the learner to construct a labeled data point [14]. Finally, pool-based querying takes the data that is highly anticipated to be of a certain class and that has a certain threshold of informativeness and transfers it, and its label, to the training data [15]. Each query type has an associated algorithm to optimize accuracy; for example, with pool-based querying every time data points are transferred to the training data, the learning model is retrained [13]. This iterative process is terminated once a specific criterion is met.

Informativeness Calculation

Least Confidence

With this strategy, informativeness is based on the confidence a classifier has regarding each data point. Data points that are more likely to be accurately labeled will have a higher informativeness [13].

Smallest Margin

This strategy seeks to be more data inclusive by basing informativeness on the difference between the top two label probabilities. The smaller the difference, the more informative a data point is [15].

Maximum Entropy

This strategy utilizes the entropy formula for each data point and the probabilities of their labels. The larger the result, the more informative the data point is [13].

1.5.13 -- Survey Review for Academic Performance Prediction

Educational data mining is often aided by databases of academic institutes, and online databases [16]. More specifically, the student information stored by universities such as GPA, course grades, demographics, and more offers substantial information that is capable of providing the information necessary for a successful student performance prediction model. Furthermore, according to Joosten and Cusatis, not much is known between the relationship of demographic data and academic expectations [17]. Aside from academic databases, the other option to provide adequate educational data for machine learning use is the implementation of student surveys.

Jootsen and Cusatis Experimentation Analysis

A study performed by Joosten and Cusatis sought to highlight the key characteristics, obtained through a survey, that are quality indicators of a student's success [17]. This study examines the following:

Learner Support – Measurement of student perception of course expectations, policies, direction clarity, accessibility of the instructor, and accessibility of course materials. [7]

Design and Organization – Measurement of student perception of the course outline with learning objectives. More specifically, this focused on the work given to students and its quality [7]

Content – Measurement of student perception of the material provided for them such as software, textbook, and other miscellaneous resources [7]

Interactivity with instructor – Measurement of student perception of the quality of interactivity provided by their instructor in the form of answering questions, timely responses, etc. [7]

Interactivity with peers – Measurement of student perception of course's incentive to interact with their classmates [7]

Assessment – Measurement of student perception of grading quality.

Full instructional characteristics – Conglomerative measure of the preceding characteristics in one measurement [7]

Disability – Boolean variable representing a disability present

First-generation status – Variable representing the highest-level of education obtained by a student's parents [17]

Minority status – Variable representing the minority status of American Indian, Asian American, African American, Native Hawaiian, White, or two or more races.

Low-income status – Boolean variable representing whether the student is eligible for the Pell Grant or otherwise [17]

It should be noted that this studies' definition of successful student performance was measured by the self-reported perception of knowledge gained, the satisfaction of each student in regards to their course and its material, and the academic performance measured by letter grade [17]. The results of this experimentation concluded that, in order of correlation, design and organization, learner support, student interaction with instructor, content design and delivery, and assessment are all significant metrics in successfully predicting student success [17]. However, it is postulated that the conglomeration of all metrics recorded are the biggest aid in performance modeling.

Fatima, Siddiqui, and Arain Research Analysis

Research performed by Fatima, Siddiqui, and Arain sought to find correlation between student punctuality and parent participation with student performance [16]. It should be noted that the population for this experimentation consisted of those still attending primary school, that is grades 1-12, in an online learning enviornment. Nonetheless, useful metrics in predicting student performance in this scenario may still be helpful in a college academic success model. Metrics considered for their machine learning model are as follows:

Gender – Boolean value of the student's gender [6]

Country – The country a student belongs to [6]

Birthplace – The place of birth for a given student [6]

Parent Responsible – Boolean value representing mom or dad as the responsible parent of the student [6]

Levels of Education – 3 option metric representing the educational stage of the student (high, medium, low) [6]

Student Grade – Grade level of student [6]

ID of Section – Class section assigned to respective student [6]

Student Semester – Semester the student is in [6]

Course – Offered course respective student is attending [6]

Punctuality of student in the class – Number of days the student is in class [6]

Parent involvement – Boolean value representing whether the parent has completed a form [6]

Satisfaction of Parent – Positive or negative outlook parent has with student's curriculum [6]

Group Discussion – Student interaction during group discussions [6]

Resources visited by a student – No description offered

Raising hands – How often the student raises their hand [6]

Assignments viewed by a student – Number of assignments viewed by student [6]

Results of this experimentation conclude that student absence provides the strongest correlation to academic performance [6]. Other promising metrics that could apply in a college setting are raised hands, visited resources, viewing assignments, and discussion groups with feature scores of 0.152, 0.129, 0.11, and 0.091 respectively.

Chapter 2

Prior Research Done

"Student Performance Prediction Model Based on Lion-Wolf Neural Network"

2.1 -- *Problem Definition*

The goal of this paper's research is to predict the semester marks of separate semesters for students
based on varying influencing factors. Feature selection and importance is based on entropy with minimal entropy being best. Lion-wolf training is used for determining neurode weights [2].

2.2 -- *Data Collection*

Data surrounding student individuality, environment, schooling, and family is collected. This was
recorded through a questionnaire. Furthermore, HSC and SSLC scores are also taken into consideration.
Parents of the students are also questioned for data collection [2].

2.3 -- *Data Selection*

In order to reduce complexity, data selection is used. An entropy function is used to measure a feature's
importance. Entropy is the measure of uncertainty of a random variable [2].

2.4 -- *Prediction*

The selected features from above will be used as inputs to the supervised neural network through collected semester marks. The predicted outputs of the neural network include 8 semesters [2].

2.5 -- Results and Discussion

The Lion-Wolf algorithm outperformed Lion, Grey Wolf Optimization Algorithm, and the Genetic Algorithm for a variety of training data in terms of mean square error and root mean

square error. Additionally, the Lion-Wolf algorithm outperformed the others for a variety of input layers as well. Furthermore, it's seen that the algorithm performs better with a decreased number of hidden layers [2].

2.6 -- *Critique of Paper*

This research is commendable because as stated in its literature review, other experiments with hybridization resulted in unfavorable results. By blending the strengths of the Lion and Grey Wolf algorithms, this research concluded in results better than a variety of other algorithms. For this reason, the Lion-Wolf algorithm seems to be a good approach for building a neural network in terms of EDM.

Furthermore, the variety of factors (individual, environmental, family, and schooling) most likely also were a great reason for this algorithm's success in addition to the feature selection used.

Admittedly, this novel research paper was an inspiration for the project surrounding this thesis. The authors of this paper were able to prove student prediction is possible despite limited datasets associated with educational data mining.

Chapter 3

Methodology

3.1 -- Data Collected

In order to predict student drop likelihood, a neural network is used to predict the likelihood of a student dropping a course with a "W". A mixture of quantitative and qualitative data is collected for the data input. Quantitative data is used to gage an objective perspective of how a student is performing academically. To supplement this, qualitative data is also collected to get the student's subjective perception of the class and the outcome of their performance.

3.2 -- Survey

Collected questions are obtained using a google forms survey that is primarily sent out from course instructors. Incentive to participate in the survey is extra credit in the course. The courses that participated in this data collection are ENGI 1331 and COSC 2430. Furthermore, the undergraduate engineering college as a whole received the option to participate in the survey as well. Lastly, another means of survey distribution is through the LAUNCH tutoring center. Surveys provided to tutorees were in the form of electronic mail. None of the questions asked put the participants at risk. The total population that participated in this study is 50 students. The number of students from each college who took part in the survey can be seen in the table below.

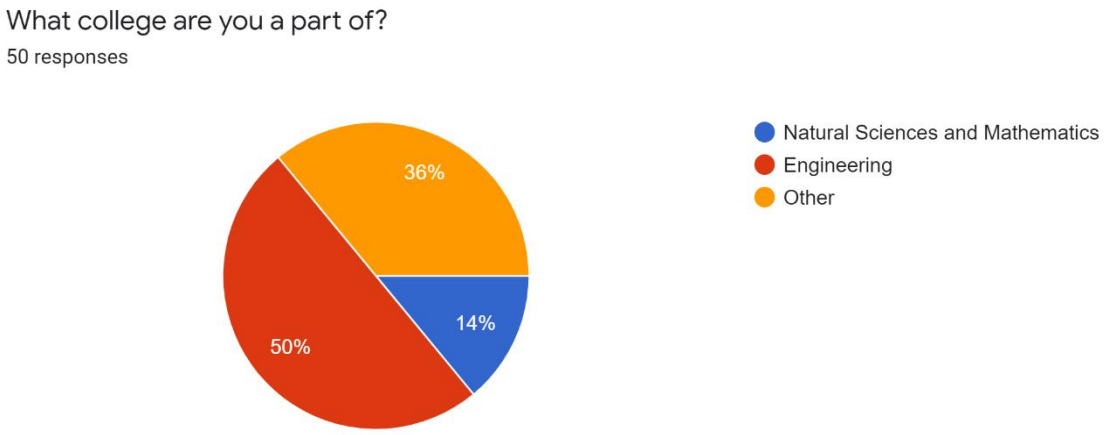

Figure 5: Breakdown of College Survey Participation

Design of the survey questions is based on the literature review performed. The questions heavily linked to student academic performance are included in the survey. Additionally, some questions were included at the researcher discretion or recommendation of the thesis mentor. The questions that will be collected using a survey are outlined below along with possible responses and a brief description of the importance of each metric:

- What is your first and last name?
    - Free Response

    Used for identification such that UH database can be used to determine whether the student dropped the course or not without the need for a follow-up
- What is your 7-digit Peoplesoft ID number?
    - Free Response validated by numerical value containing 7 digits

    Used for identification such that UH database can be used to determine whether the student dropped the course or not without the need for a follow-up
- What college are you a part of?
    - Natural Sciences and Mathematics
    - Engineering
    - Other

Used such that duplicate optimizer can be used to determine different parameter weights for each college

- Which course are you taking this survey for?
    - List of Courses involved in Survey
    - Other

Used such that course can be an added metric representing course difficulty

- What is your gender?
    - Male
    - Female
    - Non-binary
    - Other
    - Prefer not to answer

Demographic information is recommended by UH-DIAS club and thesis mentor

- To the best of your knowledge, what is your GPA?
    - Free response validated by decimal value no greater than 4.0

Objective and quantitative data that can be used to determine student success

- How many courses have you withdrawn from with a "W"?
    - Free response validated by integer value no greater than 6

Used as additional metric to factor in withdraw decision

- To the best of your knowledge, are you eligible for the Pell Grant? (Does your household make more than $60,000 annually)
    - Yes
    - No

According to literature review, poverty is linked to more success in college courses

- Do you have a disability?
    - Yes
    - No

According to literature review, disability status is linked to decreased performance in college courses

- How often do you attend lecture?
    - More than 90% of the time
    - Between 70%-90% of the time
    - Between 50%-70% of the time
    - Between 20%-50% of the time
    - Less than 20% of the time

Class behavior information linked to student success according to literature review

- How often do you access learning material for the course?
    - Twice a week
    - Once a week
    - Biweekly
    - Almost never
    - Not applicable in my course

Class behavior information linked to student success according to literature review

- How often do you ask the instructor course related questions?
    - At least once every lecture
    - About every other lecture
    - Every now and then
    - Almost never

Class behavior information linked to student success according to literature review

- How often do you complete your assignments?
    - Always
    - Miss 1-2 assignments every semester
    - Miss 3-4 assignments every semester
    - Miss more than 5 assignments every semester

Class behavior information linked to student success according to literature review

- How early do you typically complete your assignments?
    - Very early
    - Within a few days

- Last minute
- It varies

Class behavior information linked to student success according to literature review

- How well organized do you believe your given course to be? (course outlines match learning objectives, enough time dedicated to a subject, etc)
  - Above average
  - Average
  - Below Average
  - Hardly organized at all

Course perception information linked to student success according to literature review

- How would you rate the learner support provided in terms of direction clarity, accessibility of the instructor, and accessibility of the course materials?
  - Linear scale from 1 to 10 (inclusive)

Course perception information linked to student success according to literature review

- How would you rate the quality of learning material given, such as textbook, software, or other miscellaneous resources?
  - Linear scale from 1 to 10 (inclusive)

Course perception information linked to student success according to literature review

- How would you rate the grading quality of your given course?
  - Linear scale from 1 to 10 (inclusive)

Course perception information linked to student success according to literature review

- How would you rate your performance in the course so far?
  - Linear scale from 1 to 10 (inclusive)

Self-evaluation used to aid prediction

3.4 -- Survey Translation into Array of Metrics

Each survey question will be exported to an excel file, read into a python script, and transferred into a Pandas data frame. Each row of the array represents a student's response and therefore a

single data point. Each column is a survey question that will be used as a metric in the machine learning pipeline. The student's name and peoplesoft identification number will be excluded from the data frame as they are only used as identification references such that the value of whether or not the student later dropped the course can be added as the correct output used to guide the supervised learning network. Then the data frame will be split based on the college each student attends, this metric will then be removed from the data frame. Finally, the dataset will be category encoded such that each string response is translated into an integer value. Now the data is completely numerical, contains W labels for supervised learning, and is separated by college, each can be sent to the optimizers for training and evaluation.

3.5 -- Optimizers

Optimizers used for this study are the Adagrad Optimizer, Lion Optimization Algorithm (LOA), Grey Wolf Optimizer (GWO), and Lion-Wolf Optimization Algorithm. The Adagrad Optimizer is used as the control for this research as it's well-known and well-suited for sparse datasets. The remaining three optimizers serve as independent variables because of their experimental nature. Each optimizer will be fed the same recorded EDM datasets such that they can be compared. Performance measures for each optimizer will be mean absolute error (MAE), Root Mean Squared Error (RMSE), and percent good classification.

Analysis of each optimizer is shown in Figure 4.

Algorithm	Adagrad	Lion	Grey-Wolf	Lion-Wolf
Pros	• Able to train on sparse data • Learning rate changes for each training parameter • Don't need to manually tune the learning rate	• Fast Convergence • Made for small sample sizes (50 used in research paper)	• Fast convergence • Good explorative ability	• Combines benefits of previous two • Improved performance over previous
Cons	• Computationally expensive as a need to calculate the second order derivative • The learning rate always decreasing results in slow training	• Difficult to implement	• Difficult to implement • Performance weakens the further the optimal solution is from 0	• Not great for deep neural networks consisting of more layers compared to previous two

3.6 -- Inclusion of Augmented Intelligence

While originally this research aimed to allow for expert human opinion to manually adjust weights, a new take on augmented intelligence has since replaced this original assumption.

Table 1: Optimizer Analysis

Manually adjusting weights for the algorithms proposed can have a multitude of unknown effects on the end result, potentially resulting in a significant loss in accuracy. Machine learning models, especially neural networks, have weights that are highly interconnected that don't associate to a specific metric outside of the input layer. Even the tweaking of a weight within the input layer could completely destroy the network's classification accuracy due to the weight's associations. For this reason, it is unfeasible to manually adjust weights due to the complexity associated with each model proposed.

Instead, the new take on augmented intelligence relies on the discretion of each professor. Ultimately, there are metrics a survey cannot record. Facial expressions, mental health,

economic difficulties – all are vital in determining a student's success; but, more than that, all are human characteristics difficult to measure with Boolean values.

Because the goal of this research is to predict the success of humans, it is vital to incorporate the opinions of them as well. Despite a predicted high likelihood of dropping a course, a student could have subjective views incorporated into the survey that change over time, and, in fact, be on track to earn an A instead. Conversely, a student predicted to have a low probability of dropping may start showing signs of depression and apathy. Because of human fallibility, it's important to have an expert's opinion on human behavior – another human.

Despite overwhelming success in recent years, there are still a multitude of tasks humans perform better on over artificial intelligence. Professors have collected more data on student behavior than any educational data mining project ever has. Professors document emotion, attitude, and interest of students every time they lecture or have a student in office hours. Ultimately, professors should make the call as to what intervention tactics, if any, are needed for a student predicted to drop their course. Cooperation leads to better decision making, especially that of cooperation between humanity and artificial intelligence.

3.7 -- Inclusion of Active Learning

While originally, active learning was deemed an advantageous approach to this research in terms of potential incomplete or unlabeled data taken from the survey. After receiving survey data, no incomplete or unlabeled data existed within the dataset. Furthermore, while active learning can be used to combat a limited dataset, it is not seen as advantageous to remove more data points in the dataset that has since been characteristic of overly limited. While still a potentially valuable asset to EDM with incomplete data, unlabeled data, or data that is limited within reason; for the purposes of this research, active learning has been culled from its scope.

3.8 -- Overall Structure of Machine Learning Pipeline

The structure of the machine learning pipeline is shown in Figure 6.

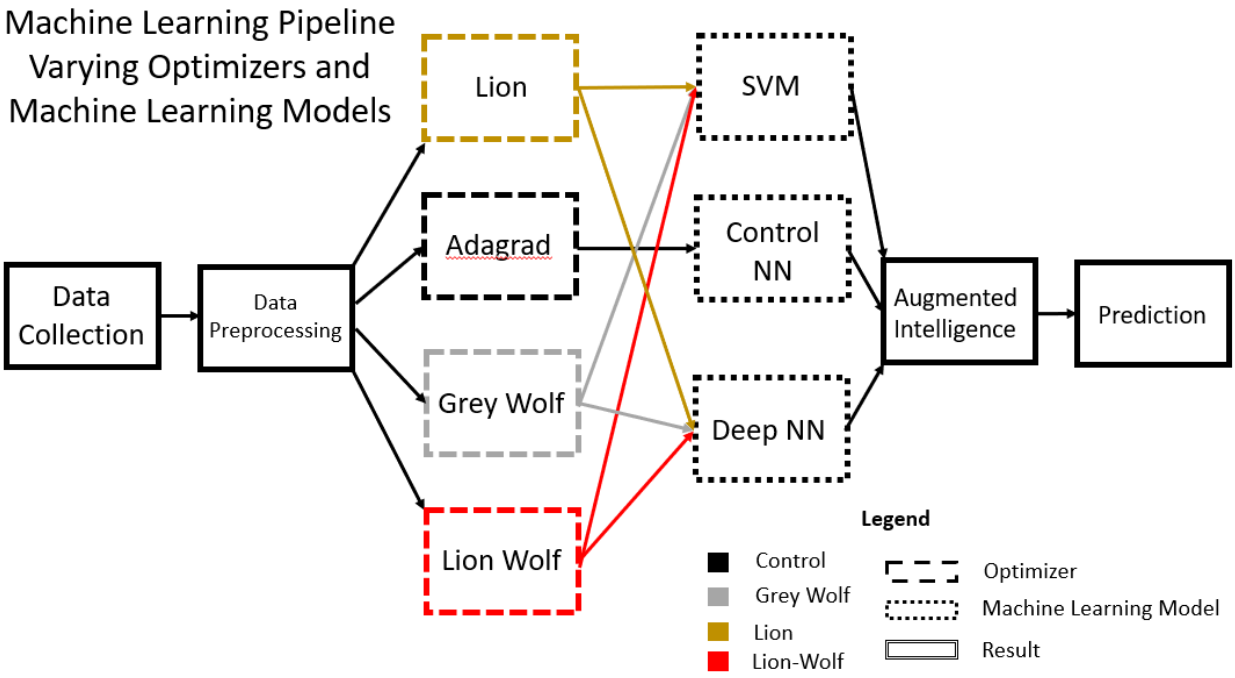

Figure 6: Machine Learning Pipeline

Chapter 4

Hypothesis

4.1 -- Optimizer Performance

While each optimizer is built for sparse data sets, a dataset of [insert population] and 17 [double check this] metrics is a difficult input for the most advanced machine learning algorithms. Based on intuition and the literature review performed, I propose the following ranking with 1 being the best-prediction results and 4 being the worst.

1. Lion-Wolf Optimizer
2. Adagrad Optimizer
3. Lion Optimization Algorithm
4. Grey Wolf Optimizer

Because the best elements of both LOA and GWO are incorporated into it's hybrid, I believe it will perform the best among the optimizers; however, I do foresee difficulty in implementing this experimental algorithm. Because of prior success with gradient descent algorithms, I foresee the Adagrad optimizer performing well due to its background and alteration to work with limited datasets. Finally, due to the multiple facets that mimic the evolutionary structure present in lions, I predict the Lion Optimization Algorithm will outperform the Grey Wolf Optimizer and its simplistic nature by comparison.

4.2 -- Heavily Linked Metrics

The metrics I predict will be most useful in determining whether or not a student will drop a course are ranked below with 1 being most useful.

1. GPA
2. Course Organization
3. W's taken
4. Learner Support
5. Grading Quality
6. Learning Material Access
7. Lecture Attendance
8. Assignment Completion
9. Assignment Completion Timeframe

10. Quality of Learning Material
11. Lecture Question Frequency
12. Quality of Interaction
13. Pell Eligibility
14. Disability Status
15. Gender

Because high GPA is a very good indicator of a student's success, I predict it will be a heavily weighted metric to determine that a student will not drop. If a GPA is relatively low, I predict other metrics will be the determining factor in prediction. Course organization is a good snapshot of the student's overall perception of a course. Low ranking will most likely indicate a student struggling in a course. W's taken will predictably play a large role in determining a student's withdrawal. Students with less W's taken can afford to drop a course and vise versa. Grading quality is a good indication of a students grades in the course as well as an indication of how they will be in the future. Lecture attendance, learning material access, assignment completion, Lecture Question Frequency, and assignment completion timeframe all indicate how much effort a student is putting into their respective course. While important, I still believe GPA, W's taken, Learner Support, and Grading Quality will still weigh more as some students can pass a course with little effort. Quality of interaction and learning material are good supplements to learner support and learning material access but will most likely not have very high weights. Finally, demographic information recorded with Pell eligibility and gender, I predict will not have as large a bearing on the prediction as indicators of student's perception of the course and student effort.

Chapter 5

Discussion

5.1 -- Neural Network using Adagrad Optimizer

This neural network was constructed using the architecture shown below.

[figure]

Adagrad, being useful for sparse datasets, was used as the optimizer for this particular network. The network was constructed over 100 epochs.

5.2 -- Support Vector Machine

The support vector machine was implemented alone using the sklearn library. To test the best kernel function when optimized under GWO and LOA, each potential kernel function and class weight was used for a single iteration to view which produced the highest accuracy. The kernel functions tested are as follows: linear, poly, rbf, and sigmoid . The two types of class weights tested were the balanced and auto parameters.

5.3 -- Naïve Bayes

The naïve bayes algorithm was implemented using the sklearn library and GaussianNB function. No hyper parameters of this function were viewed as optimizable by either the GWO or LOA; therefore, the naïve bayes algorithm was implemented individually.

5.4 -- Grey Wolf Optimizer (GWO)

The grey wolf optimizer follows the pseudocode outlined in the literature review with the exception of encirclement. The specific steps the constructed GWO uses are as follows:

I. Social Hierarchy Instantiation
    I. Construct an Alpha, Beta, Gamma, and Omega
II. Loop
    I. Calculate Fitness

i. Fitness is based on each wolf's position and fitness functions outlined below
    1. SVC's hyperparameters, regularization parameter and gamma, are assigned to the position of each wolf
    2. Deep Neural Network's number of nodes per layer (1 input, 3 hidden, 1 output) are assigned to the position of each wolf

II. Update the Pack
    i. The four positions that produced the highest calculated fitness, for every iteration, is assigned to alpha, beta, gamma, and omega respectively

III. Update Positions
    i. New positions are assigned to each wolf classification based on equations 3.3, 3.4, 3.5, and 3.6 in [8]

The Grey Wolf optimizer was fit over a series of 25 iterations for each dataset.

5.5 -- Lion Optimization Algorithm (LOA)

The Lion Optimization Algorithm follows the proposed pseudocode overviewed in the literature review. Specifically, the steps to the LOA constructed for this research are outlined below.

I. Loop
    I. Run LOA
        i. Instantiate parameters for iterations, pride number, percent of nomad lions, percent of roaming lions, mutation probability, sex rate, migration rate, maximum population, upper limit of position, lower limit of position, and dimension of population
        ii. Initialize lion population and organize them into prides leaving a few as nomads
        iii. Loop (iterations)
            1. Update best visited positions

2. Move nomad lions randomly within search space
3. Nomad lions mate
4. Nomad males attack prides
5. Female lions migrate to new pride or become nomads
6. Allocate female lions to prides
7. Kill least fit nomads
8. Calculate best visited positions

The LOA algorithm uses the same fitness calculations as that of the GWO – SVM and a deep neural network. The SVM's hyperparameters, regularization and gamma, are altered with lion's position or the deep neural networks number of nodes per layer is altered with the lion's positions.

The lion optimization algorithm is ran with a population of 50, an inner iteration count of 3, and an outer iteration count of 3 for each dataset.

5.6 -- Lion Wolf Optimizer (LWO)

The Lion Wolf Optimizer follows the structure of LOA with one exception – it's positioning function is replaced by that used in the GWO. Again, it is benchmarked using two models – SVM and a deep neural network. This optimizer is ran with a population of 50, an inner iteration count of 3, and an outer iteration count of 3 for each dataset.

5.7 -- Deep Neural Network

The deep neural network consists of three layers: input, hidden, and output. The input layer has 12 nodes and uses the rectified linear activation function. The hidden layer contains 8 nodes and uses the rectified linear activation function. Finally, the output layer contains a single node that uses the sigmoid activation function. The overall architecture of this network can be seen below:

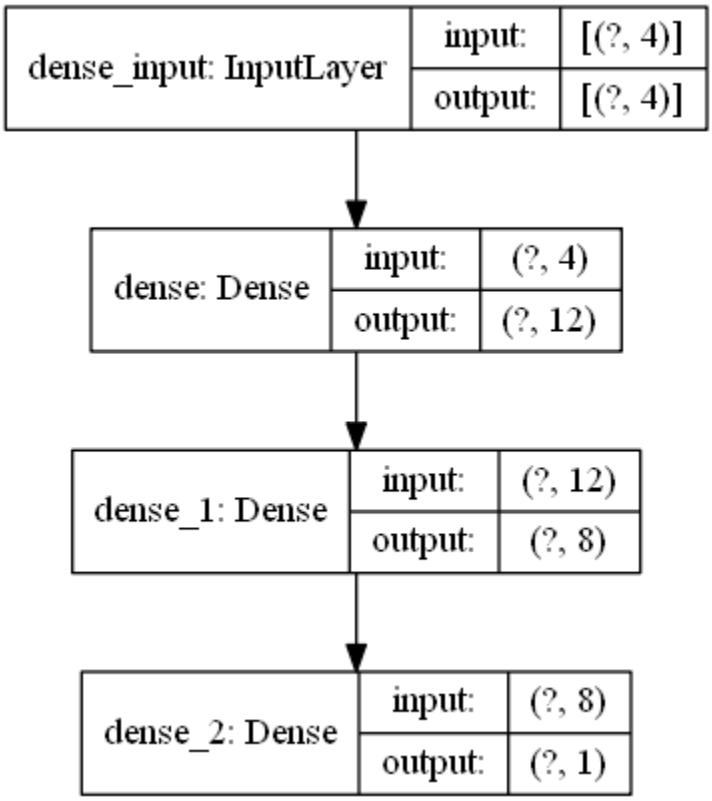

Figure 7: Default NN Architecture

Results

5.8 -- Adagrad Neural Network – Root Mean Squared Error Over Time

Iris Dataset

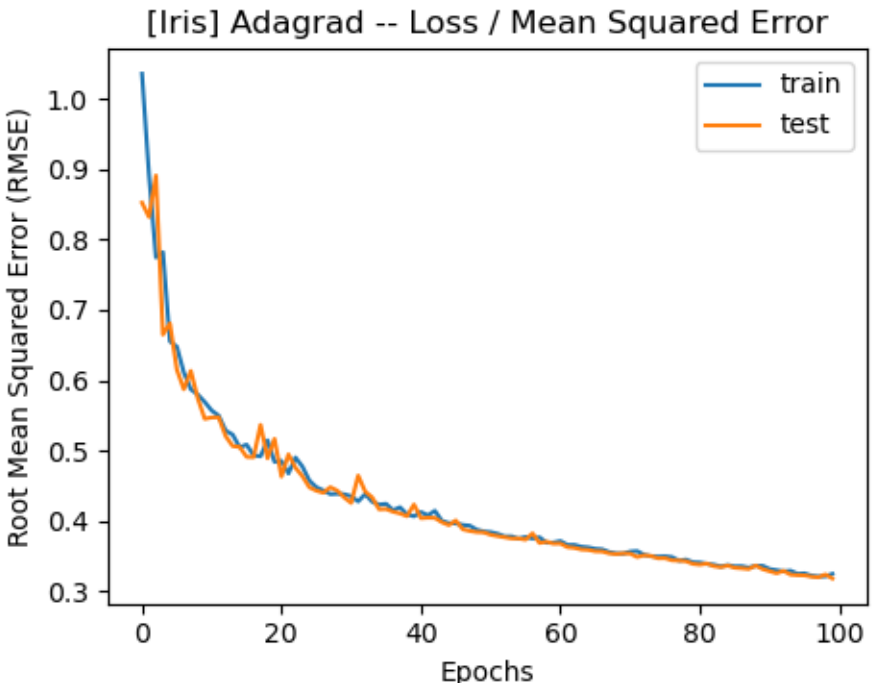

Figure 8: Iris Dataset w/ Adagrad NN -- Loss/MSE

Pseudo Survey Dataset

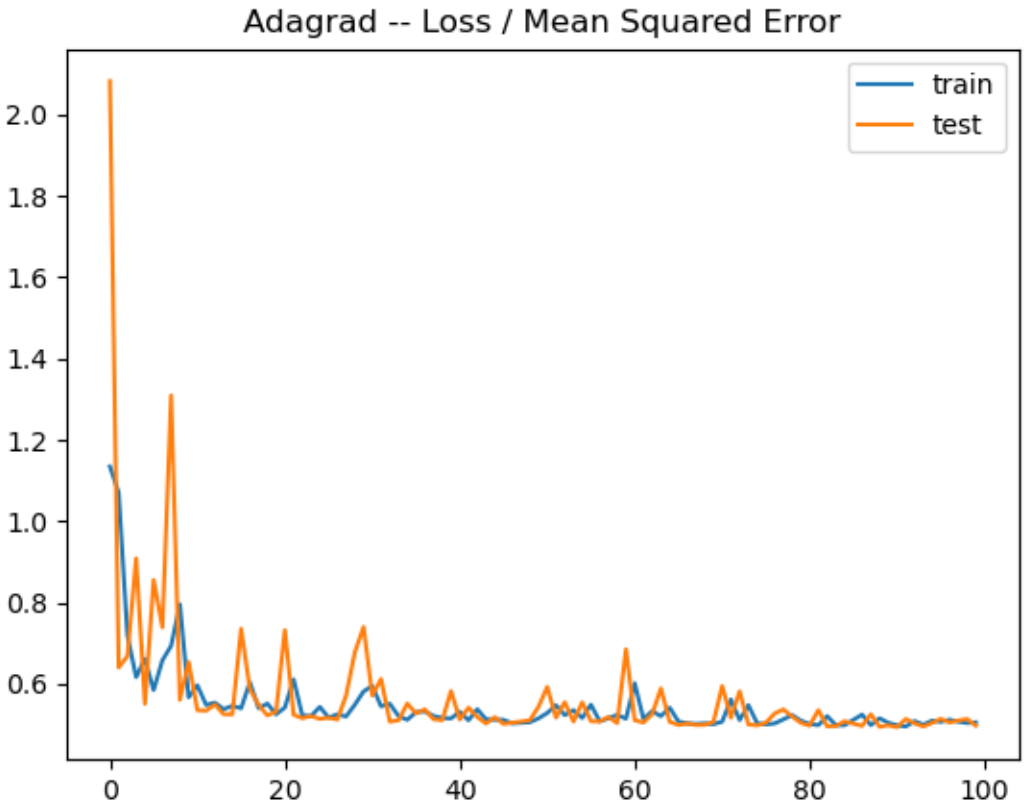

Figure 9: Pseudo Dataset w/ Adagrad NN -- Loss/MSE

Real Survey Dataset

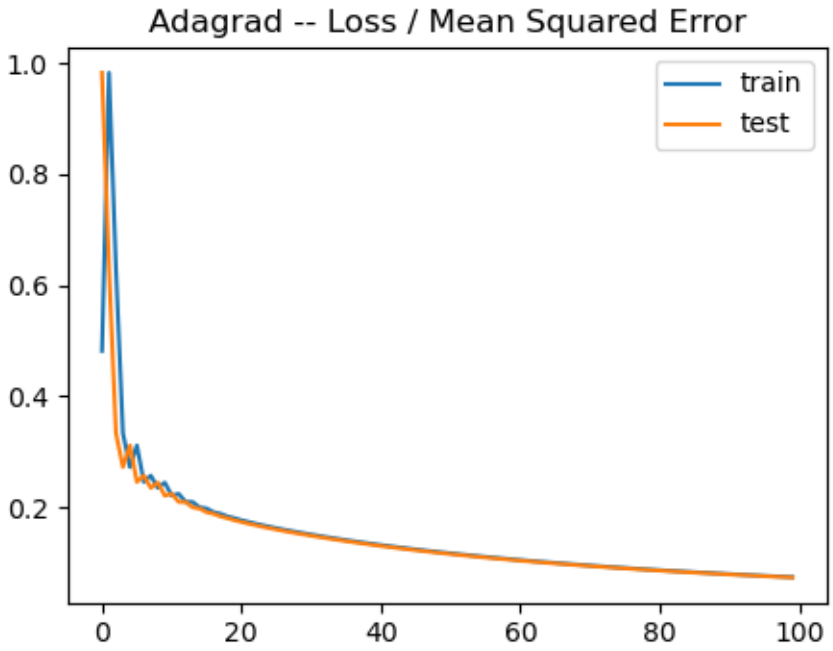

Figure 10: Real Dataset w/ Adagrad NN -- Loss/MSE

5.9 -- GWO w/ SVM Function Confusion Matrix – Testing Dataset

Iris Dataset

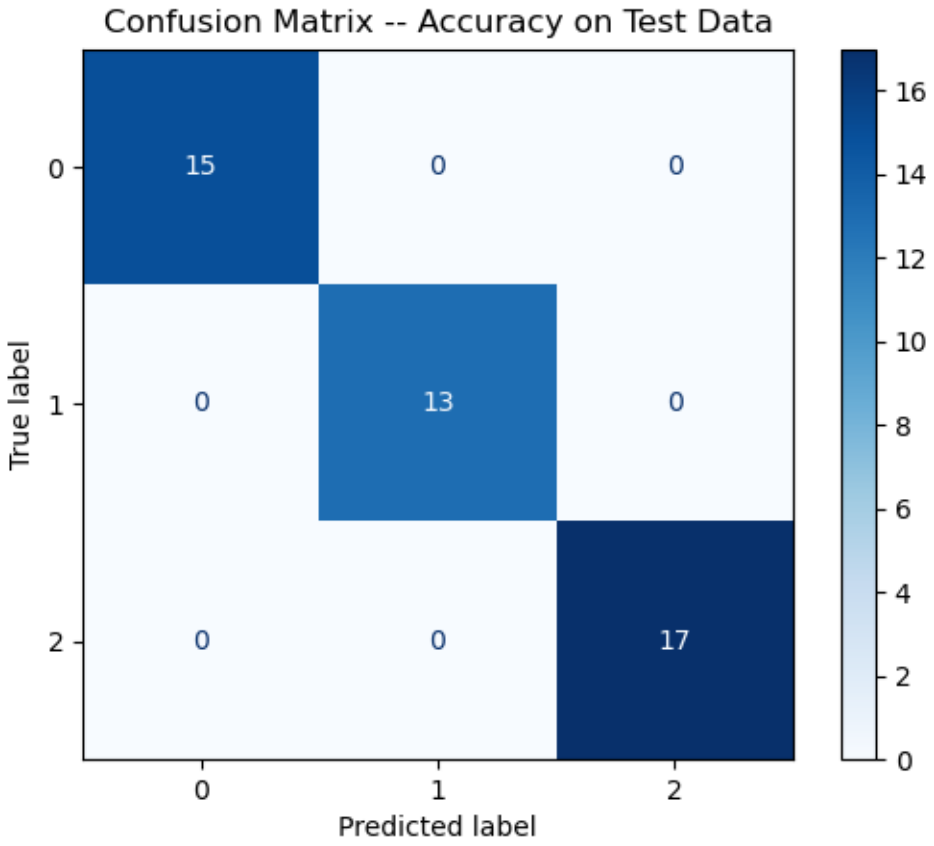

Figure 11: Iris Dataset (GWO + SVM) Confusion Matrix

Pseudo Survey Dataset

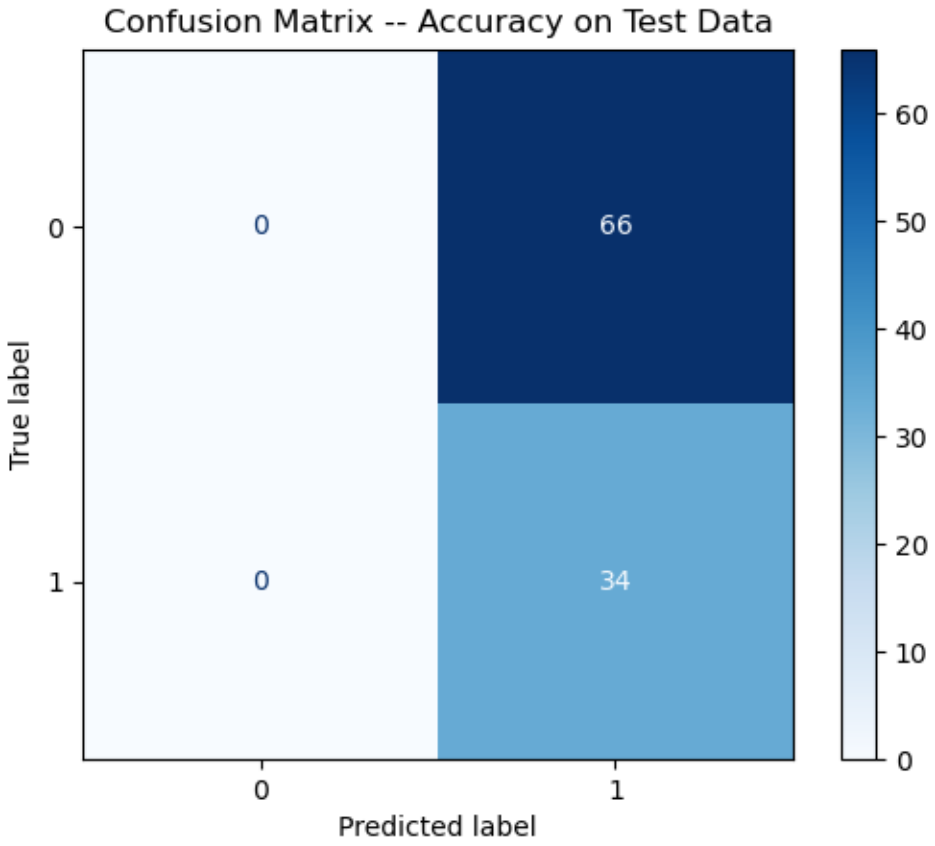

Figure 12: Pseudo Dataset (GWO + SVM) Confusion Matrix

Real Survey Dataset

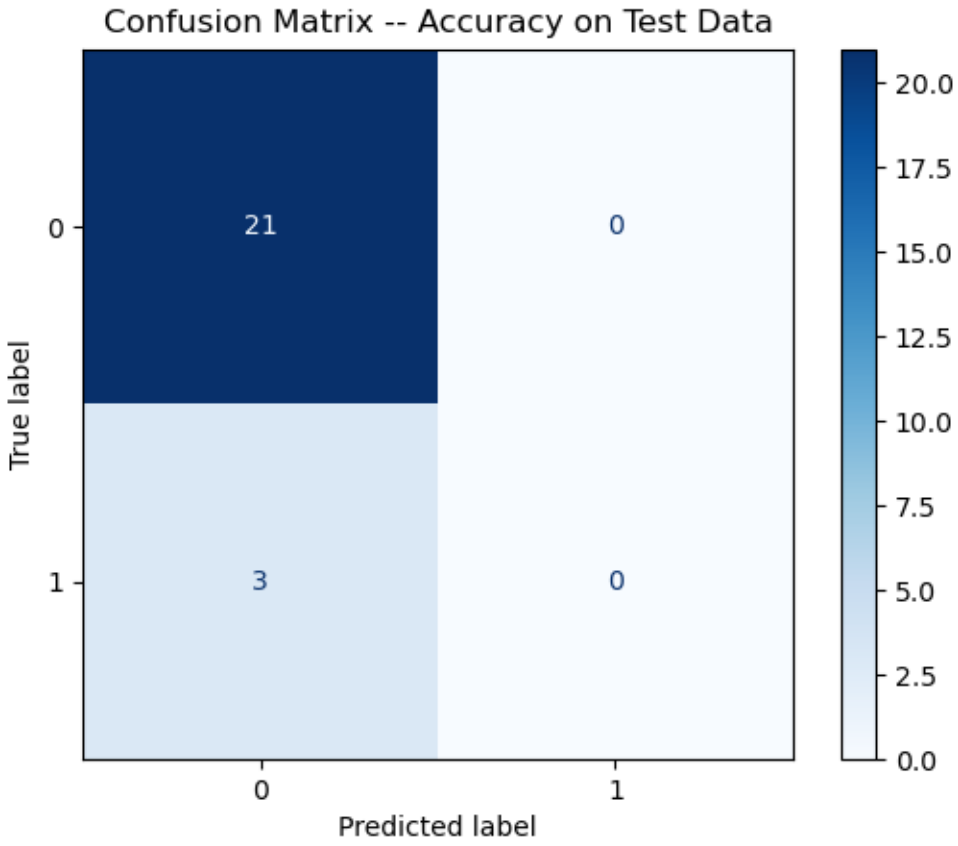

Figure 13: Real Dataset (GWO + SVM) Confusion Matrix

5.10 -- LOA w/ SVM Function Confusion Matrix – Testing Dataset

Iris Dataset

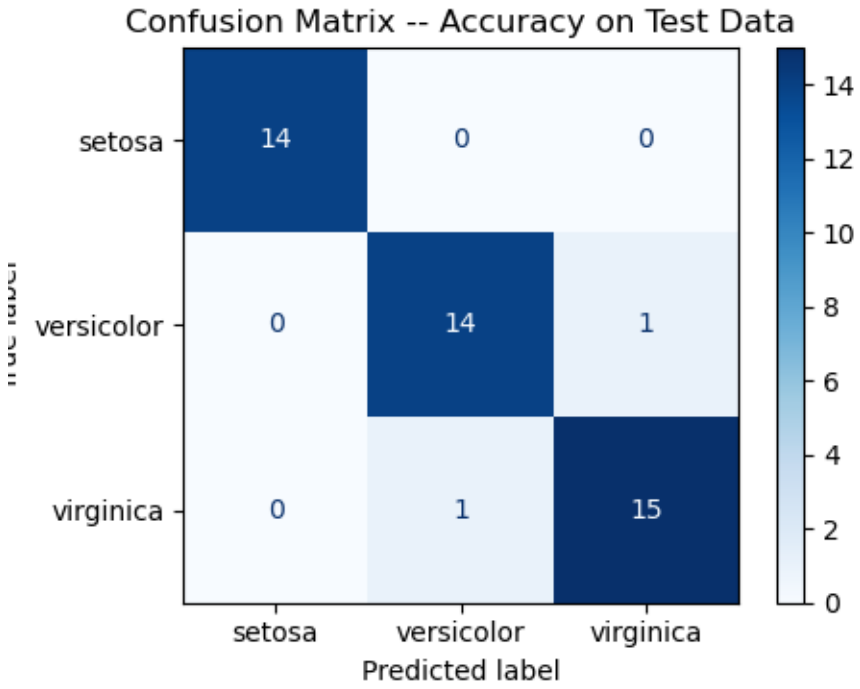

Figure 14: Iris Dataset (LOA + SVM) Confusion Matrix

Pseudo Survey Dataset

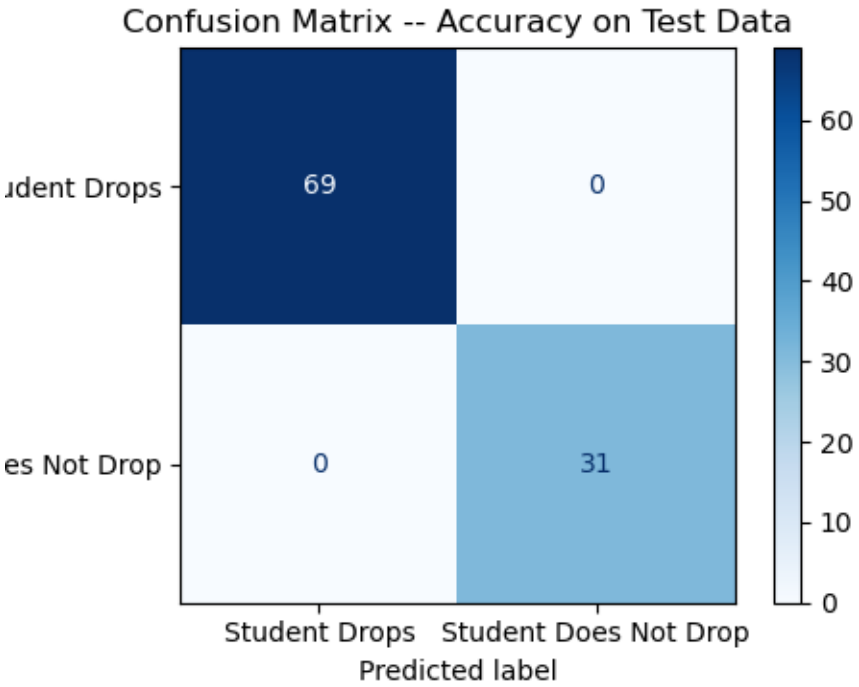

Figure 15: Pseudo Dataset (LOA + SVM) Confusion Matrix

Real Survey Dataset

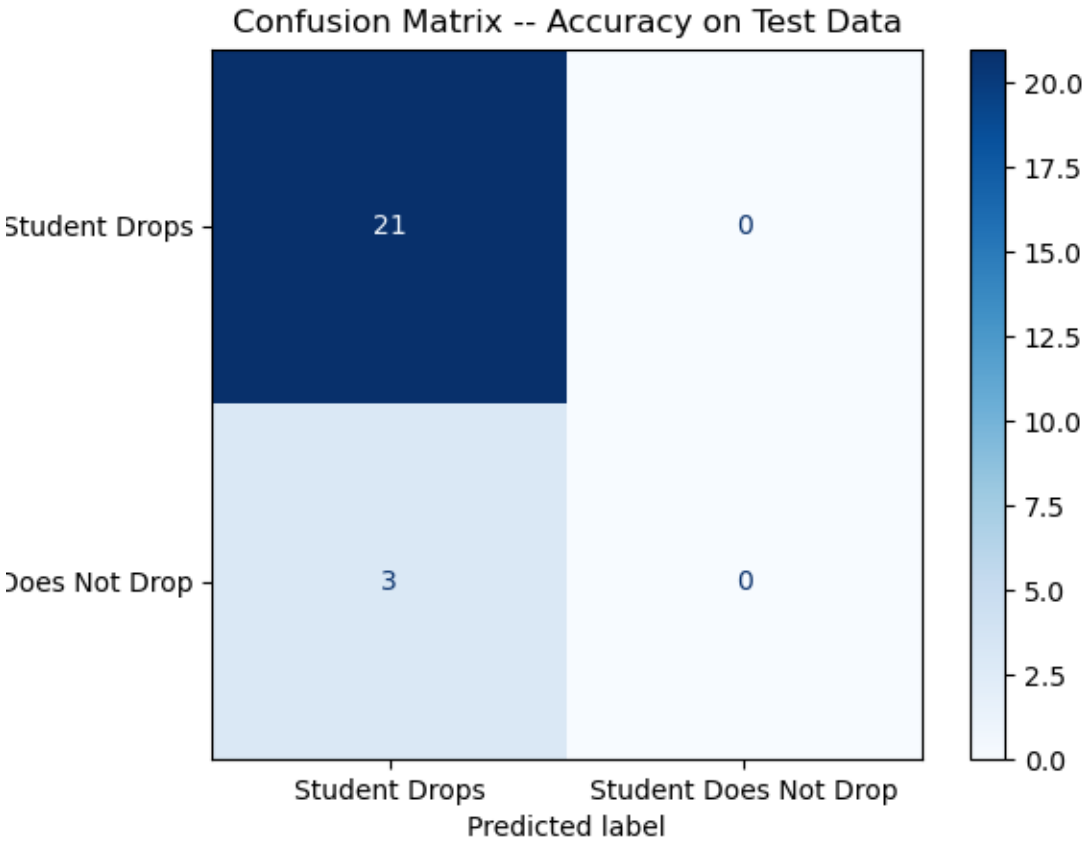

Figure 16: Real Dataset (LOA + SVM) Confusion Matrix

5.11 -- LOA w/ Deep NN Confusion Matrix – Testing Dataset

Iris Dataset

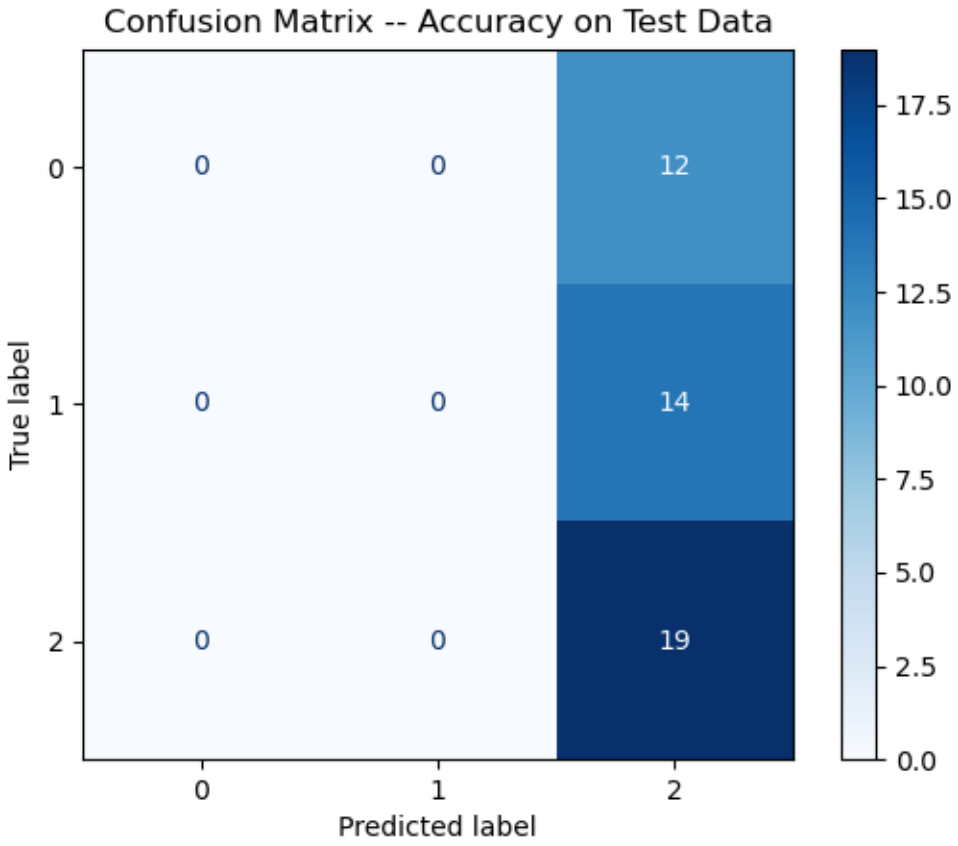

Figure 17: Iris Dataset (LOA + NN) Confusion Matrix

Pseudo Survey Dataset

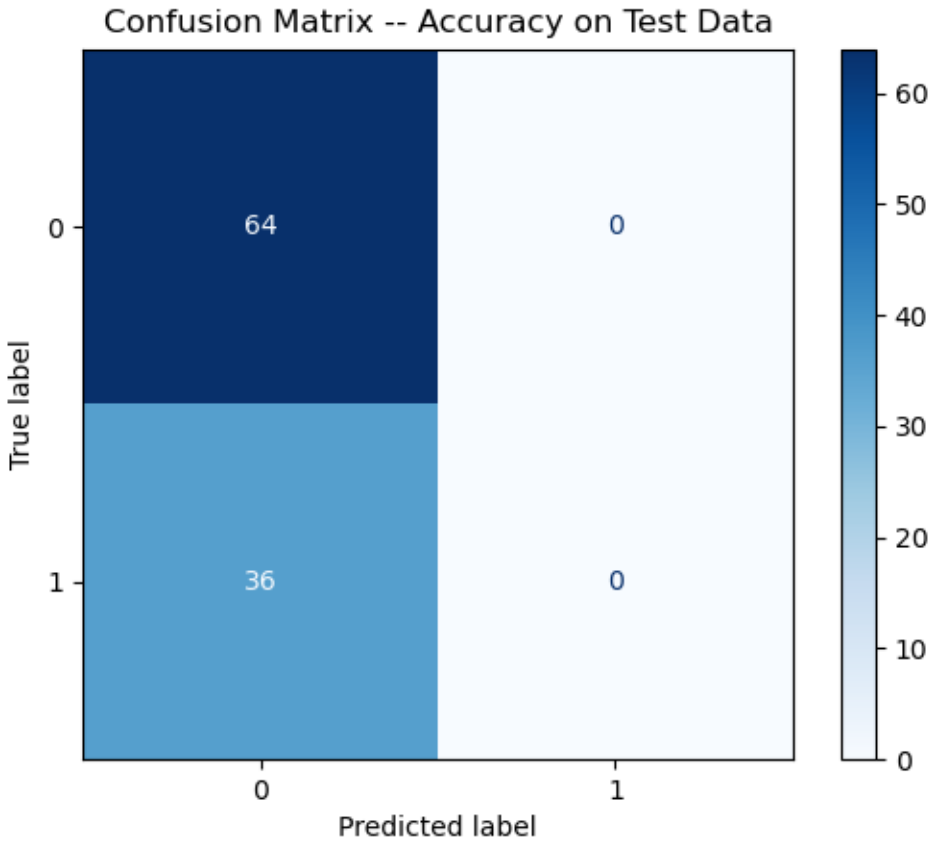

Figure 18: Pseudo Dataset (LOA + NN) Confusion Matrix

Real Survey Dataset

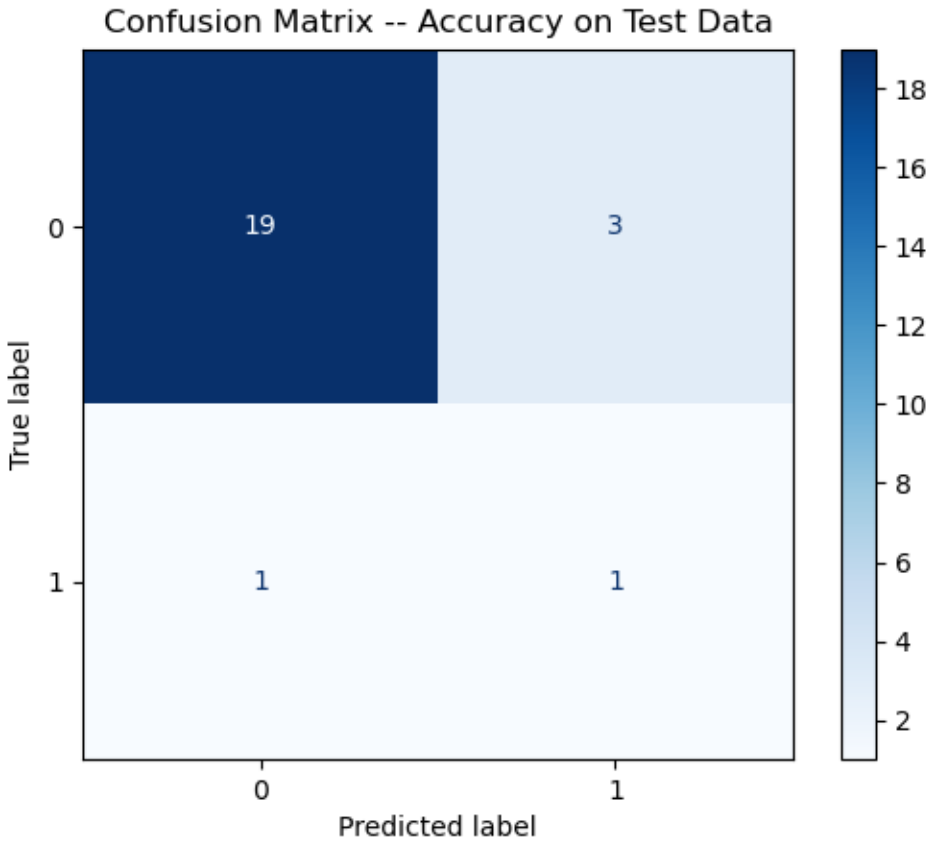

Figure 19: Real Dataset (LOA + NN) Confusion Matrix

5.11 -- Deep Neural Network – Accuracy Over Time

Iris Dataset

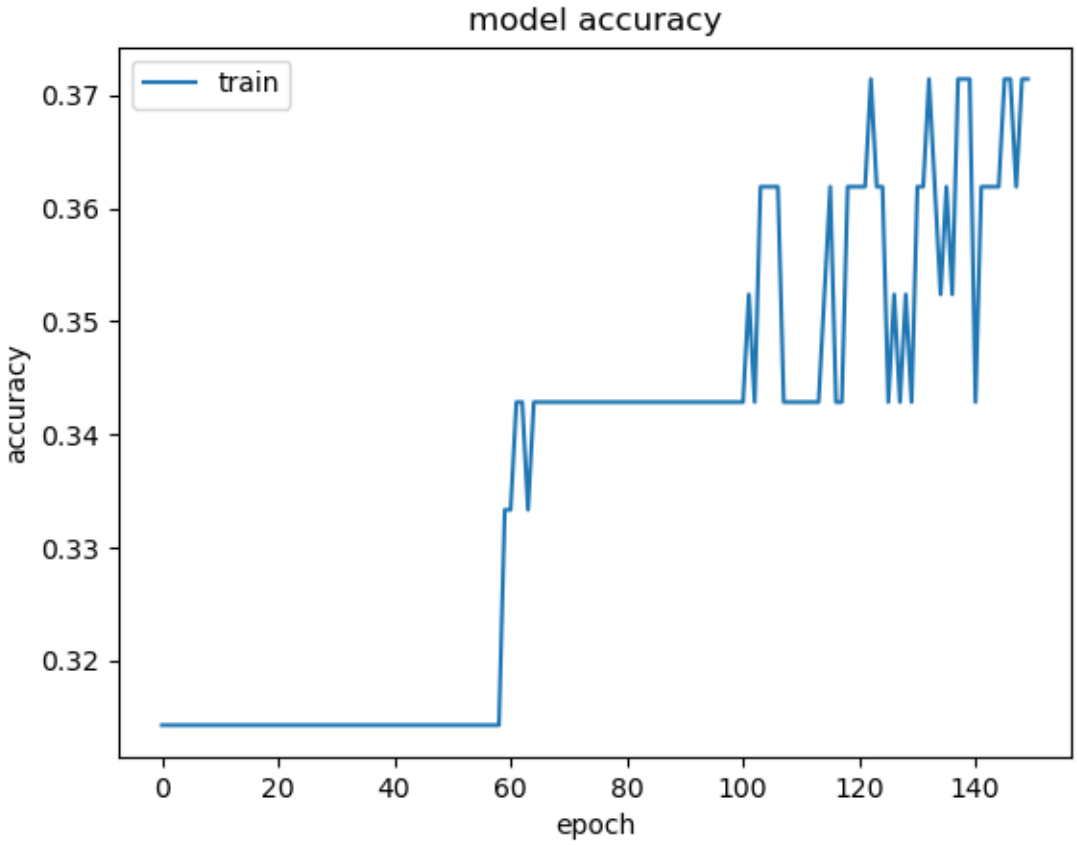

Figure 20: Iris Dataset w/ Control NN -- Accuracy Over Time

Pseudo Survey Dataset

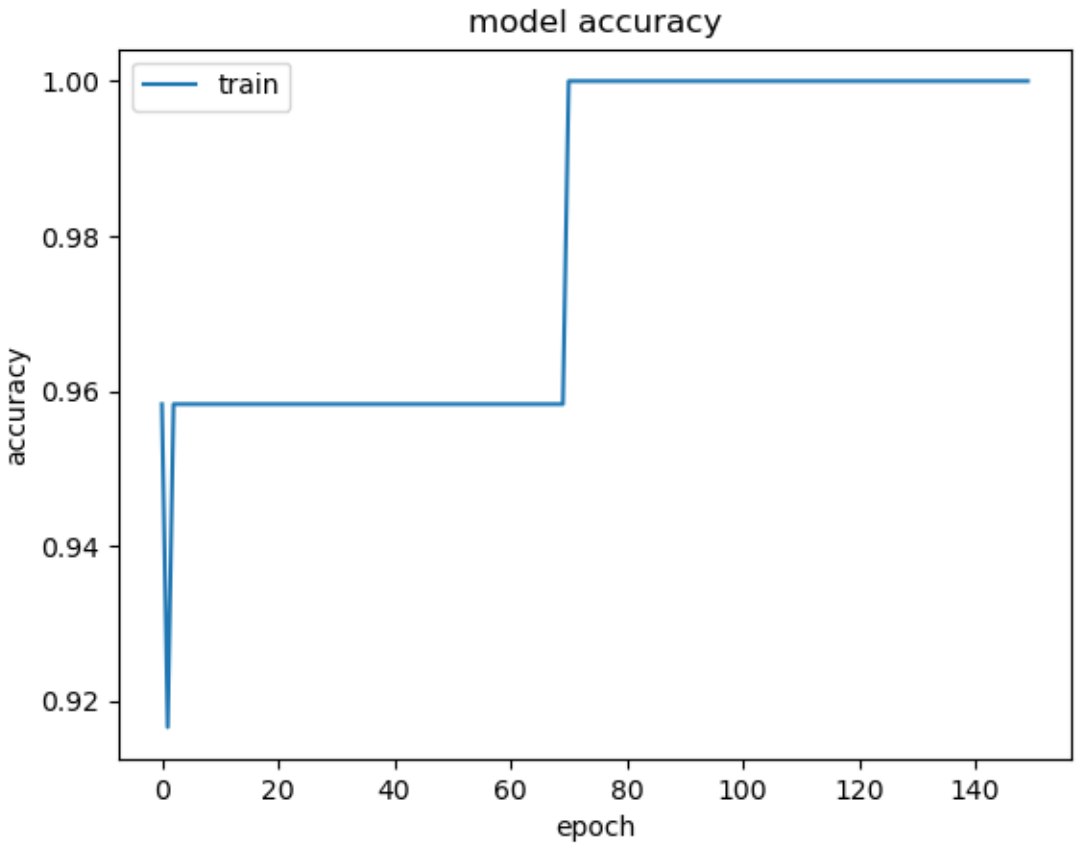

Figure 21: Pseudo Dataset w/ Control NN -- Accuracy Over Time

Real Survey Dataset

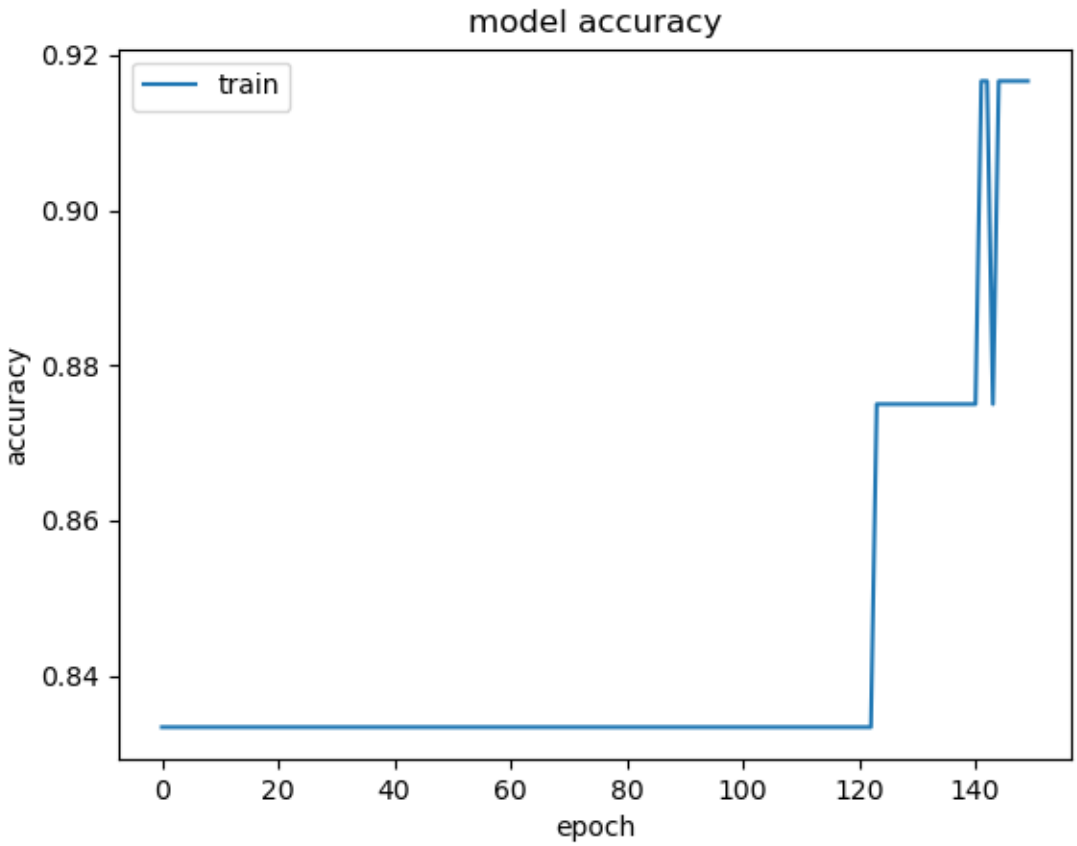

Figure 22: Real Dataset w/ Control NN -- Accuracy Over Time

5.12 -- Precision-Recall Curves – Testing Model Robustness

Pseudo Survey Dataset

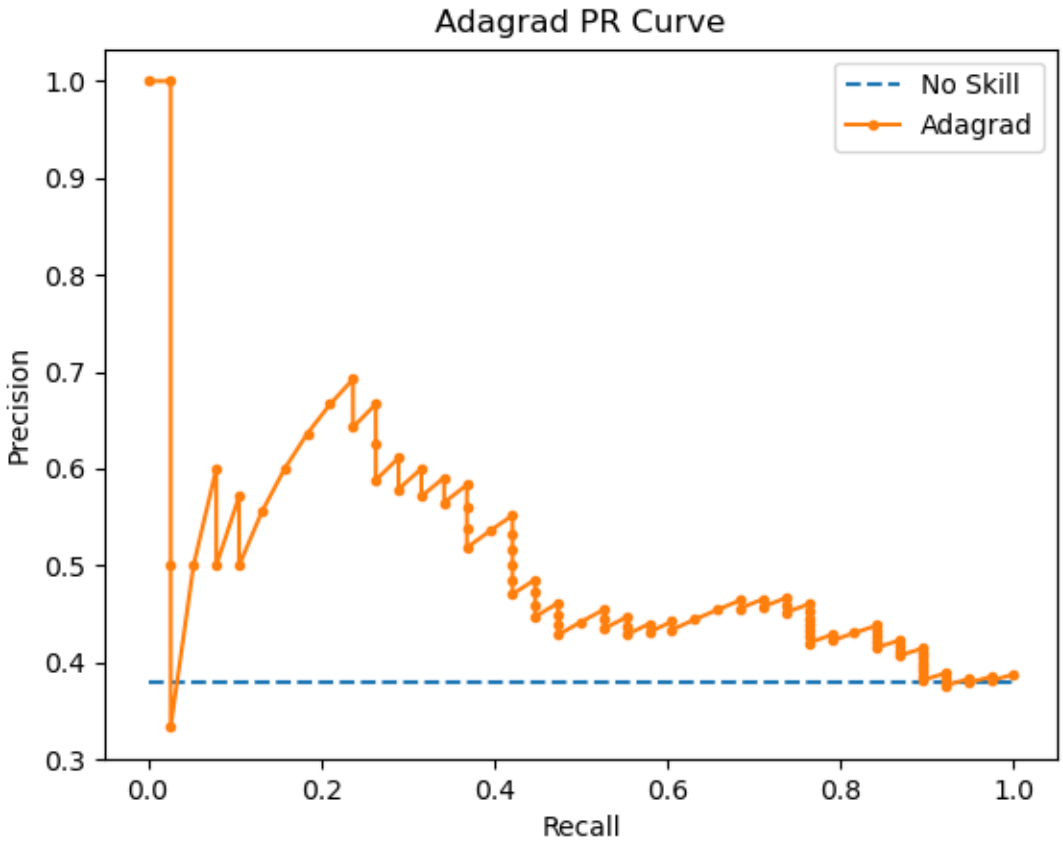

Figure 23: Adagrad -- Precision Recall Curve

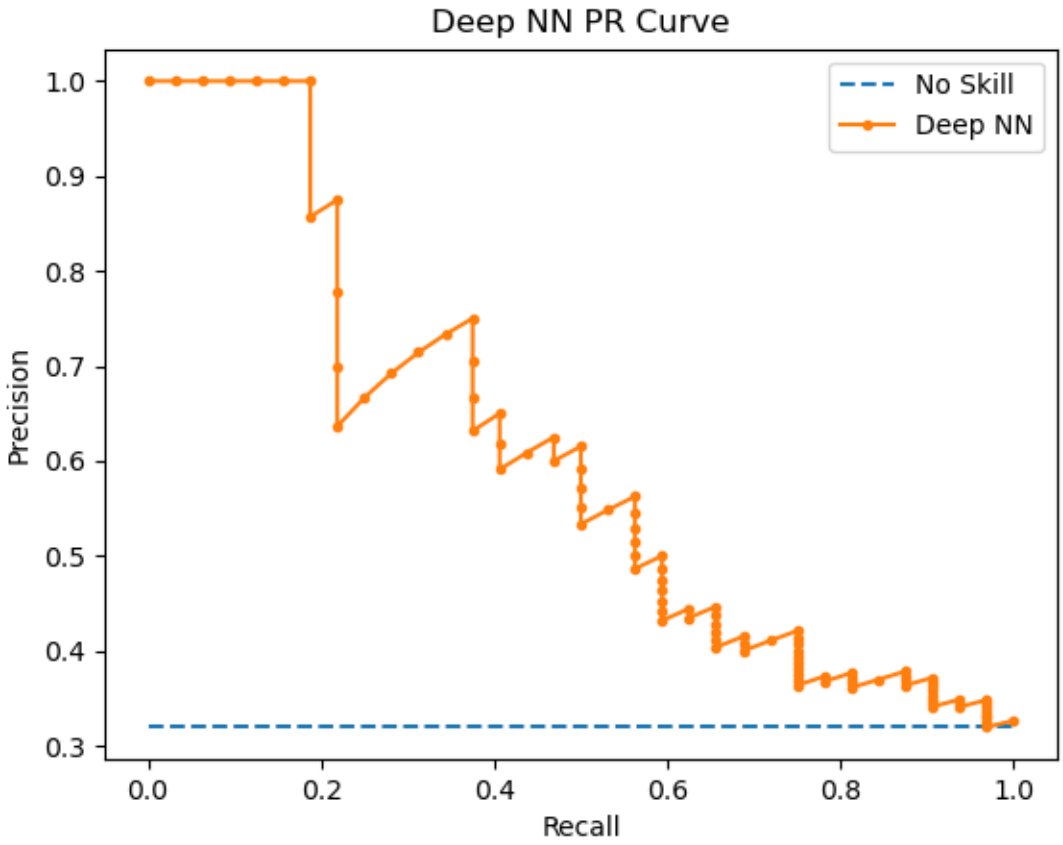

Figure 24: Deep NN Precision Recall Curve

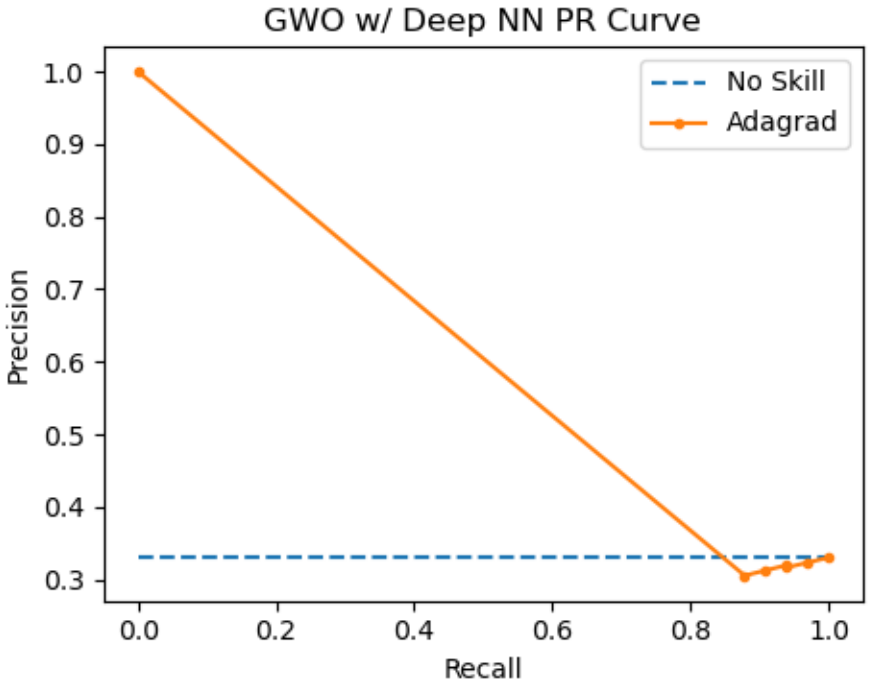

Figure 25: GWO w/ Deep NN Precision Recall Curve

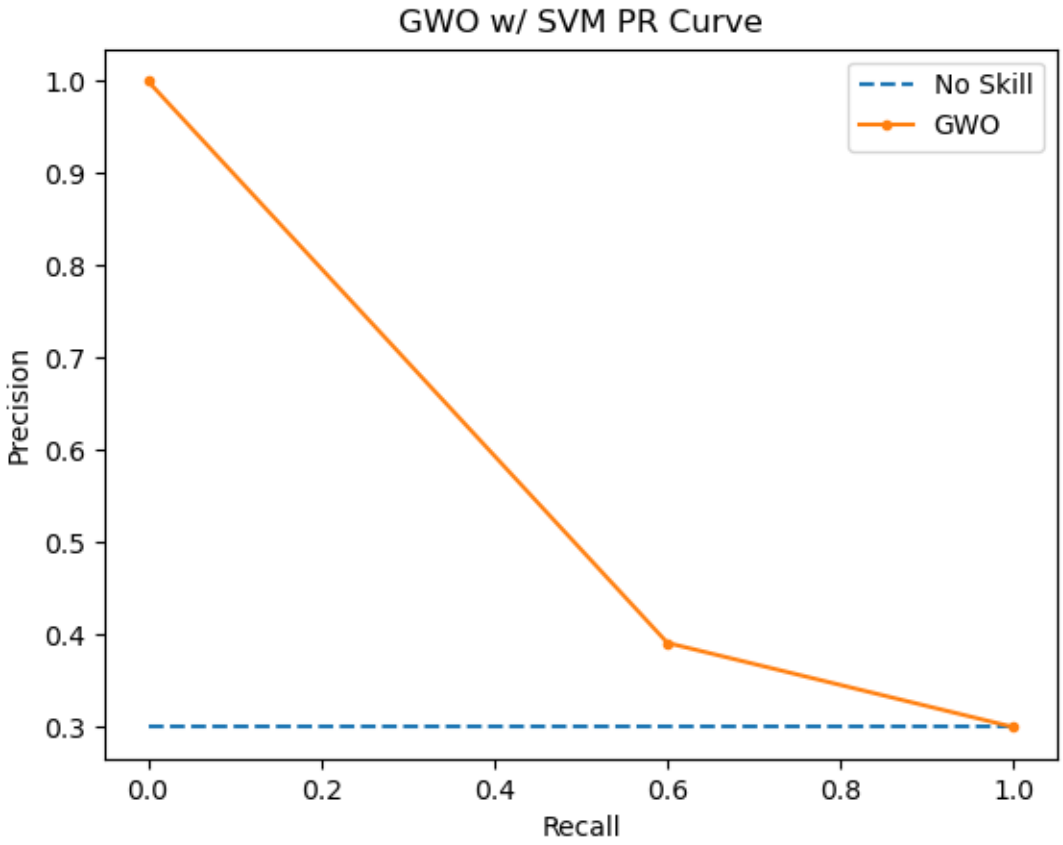

Figure 26: GWO w/ SVM Precsion Recall Curve

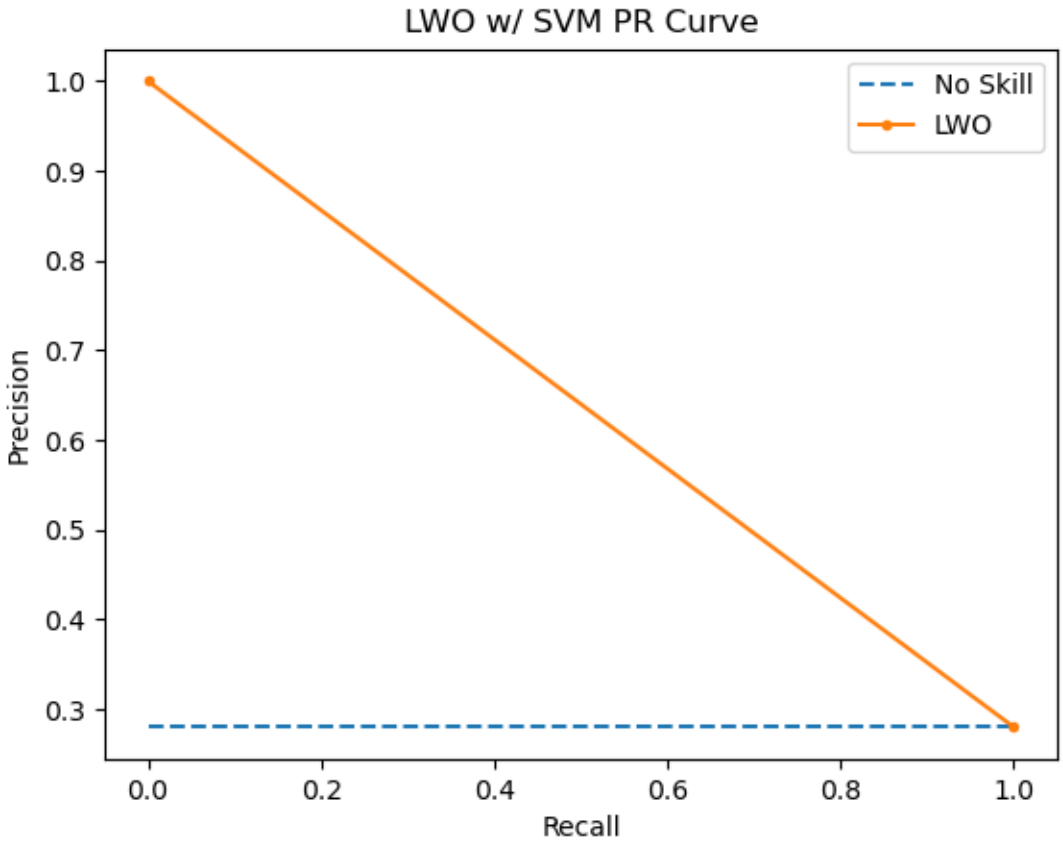

Figure 27: LWO w/ SVM Precision Curve

Real Survey Dataset

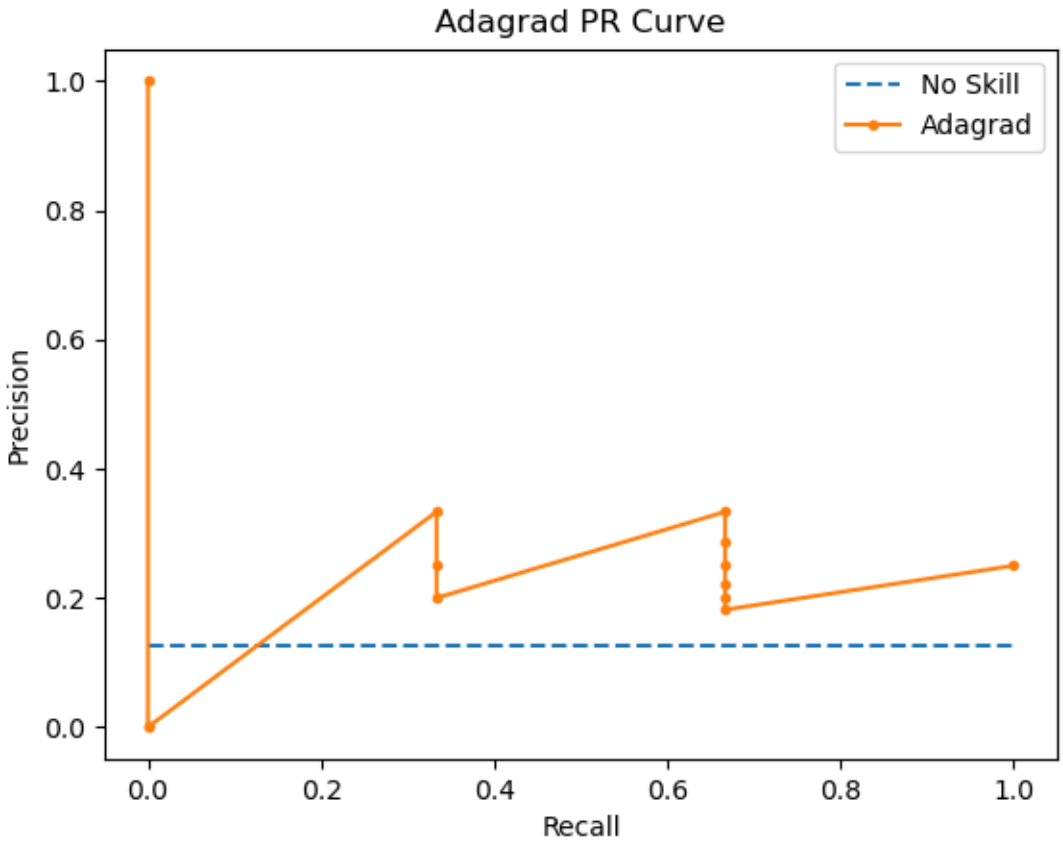

Figure 28: Adagrad Precision Recall Curve

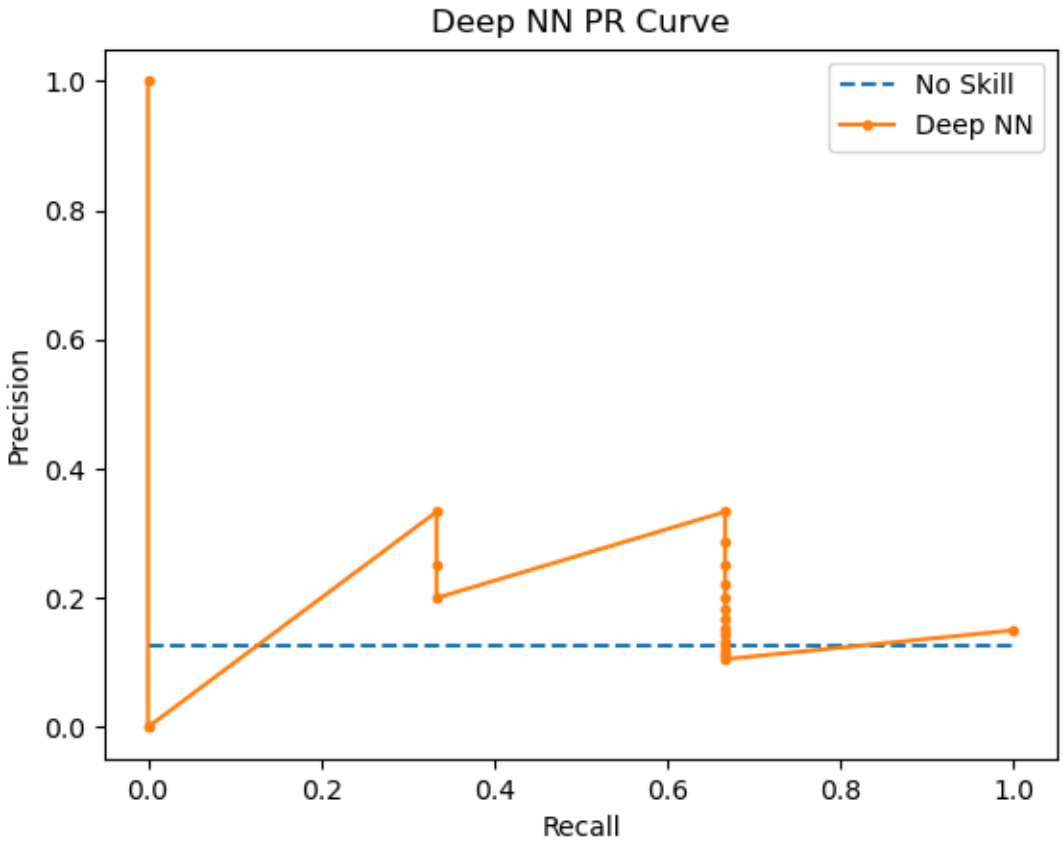

Figure 29: Deep NN Precision Recall Curve

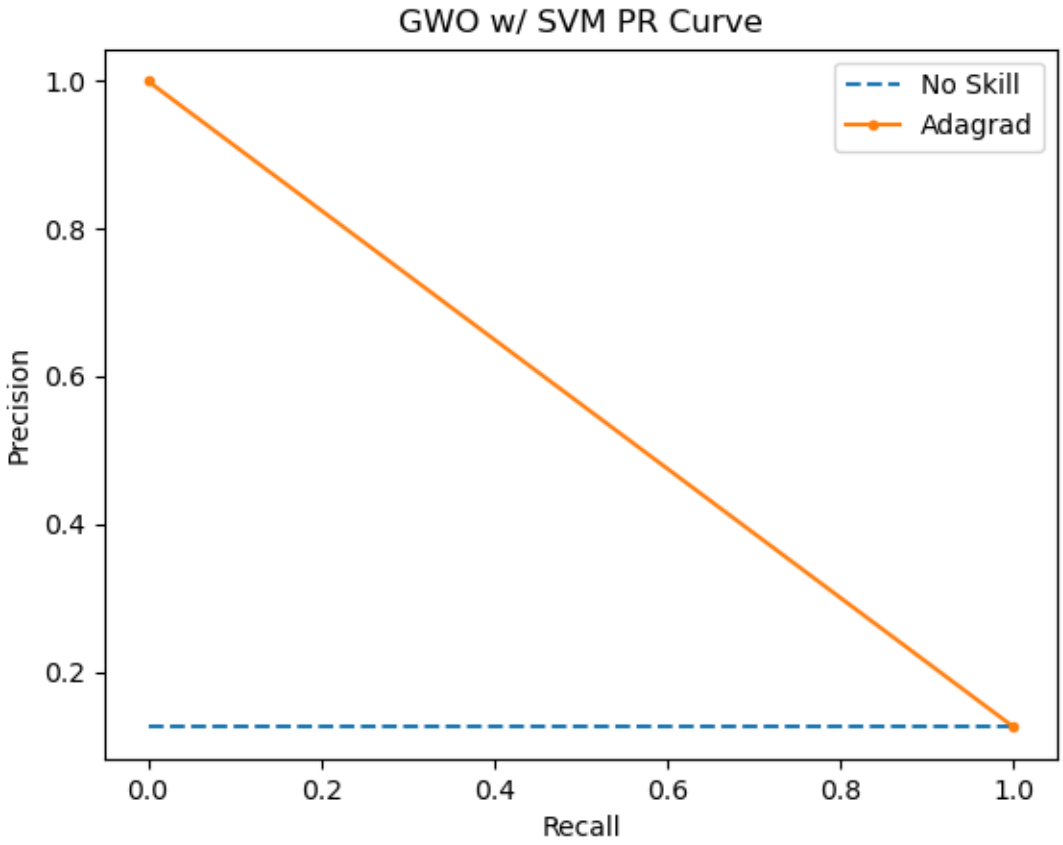

Figure 30: GWO w/ SVM Precision Recall Curve

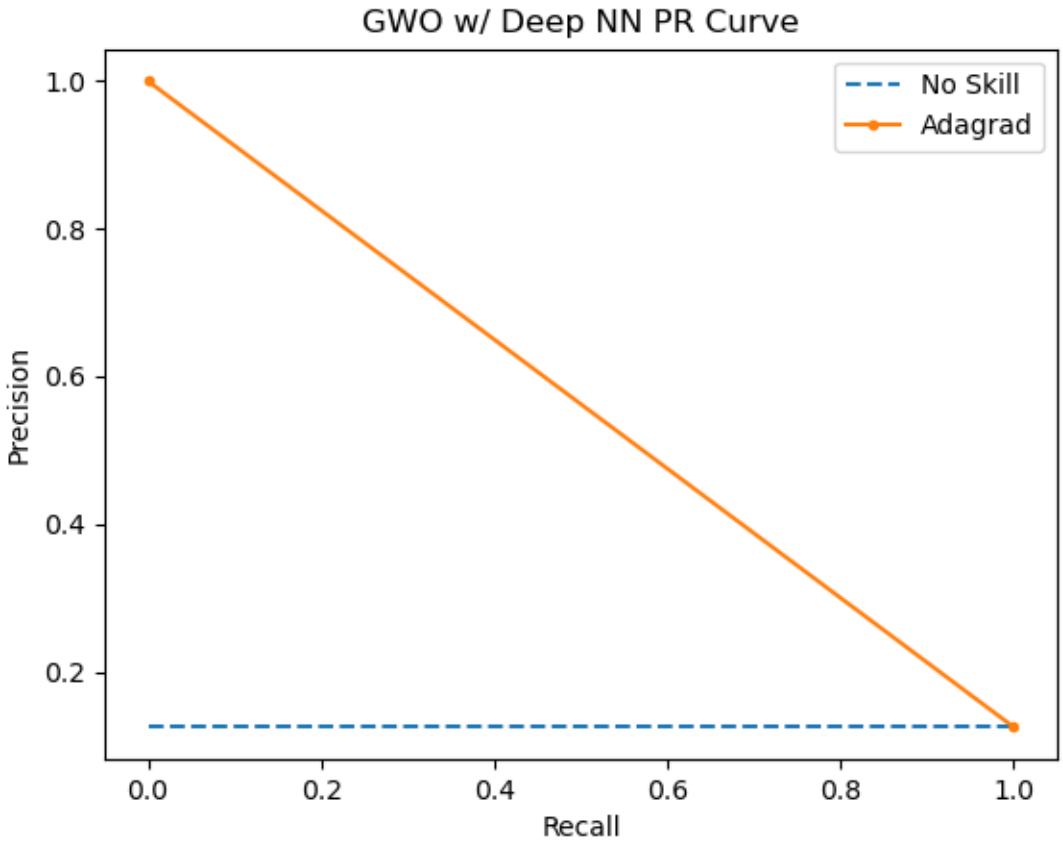

Figure 31: GWO w/ Deep NN Precision Recall curve

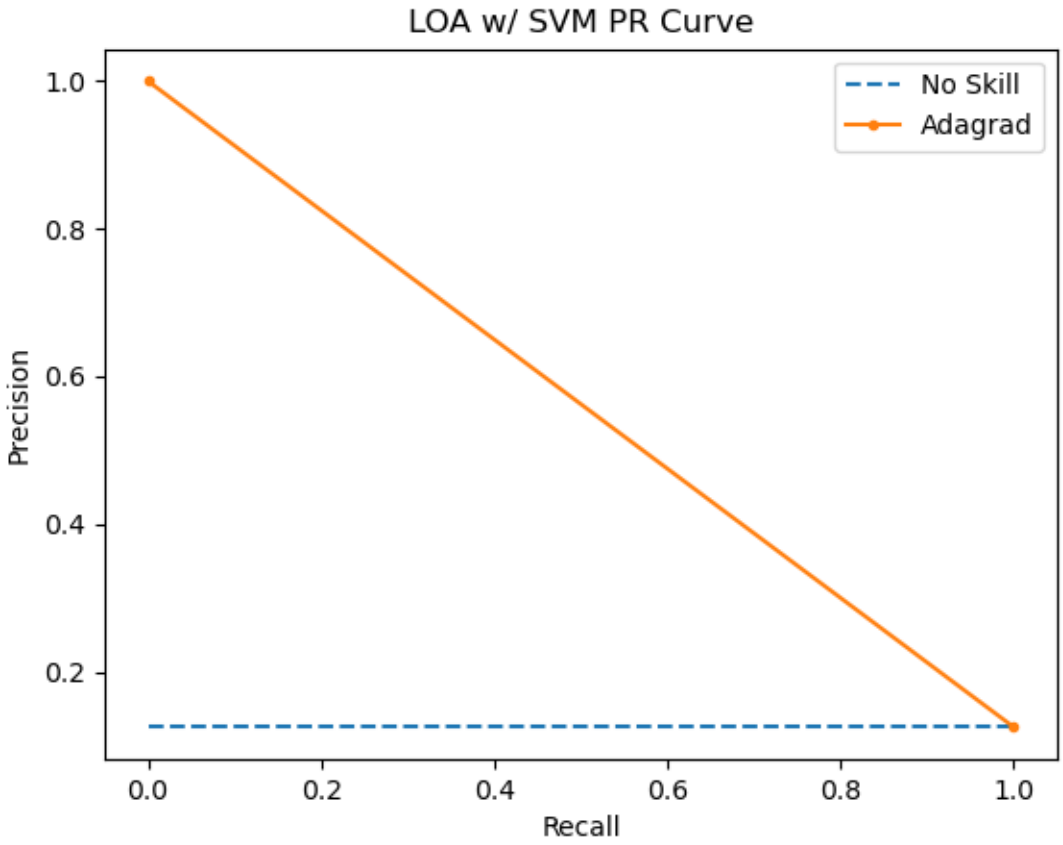

Figure 32: LOA w/ SVM Precision Recall Curve

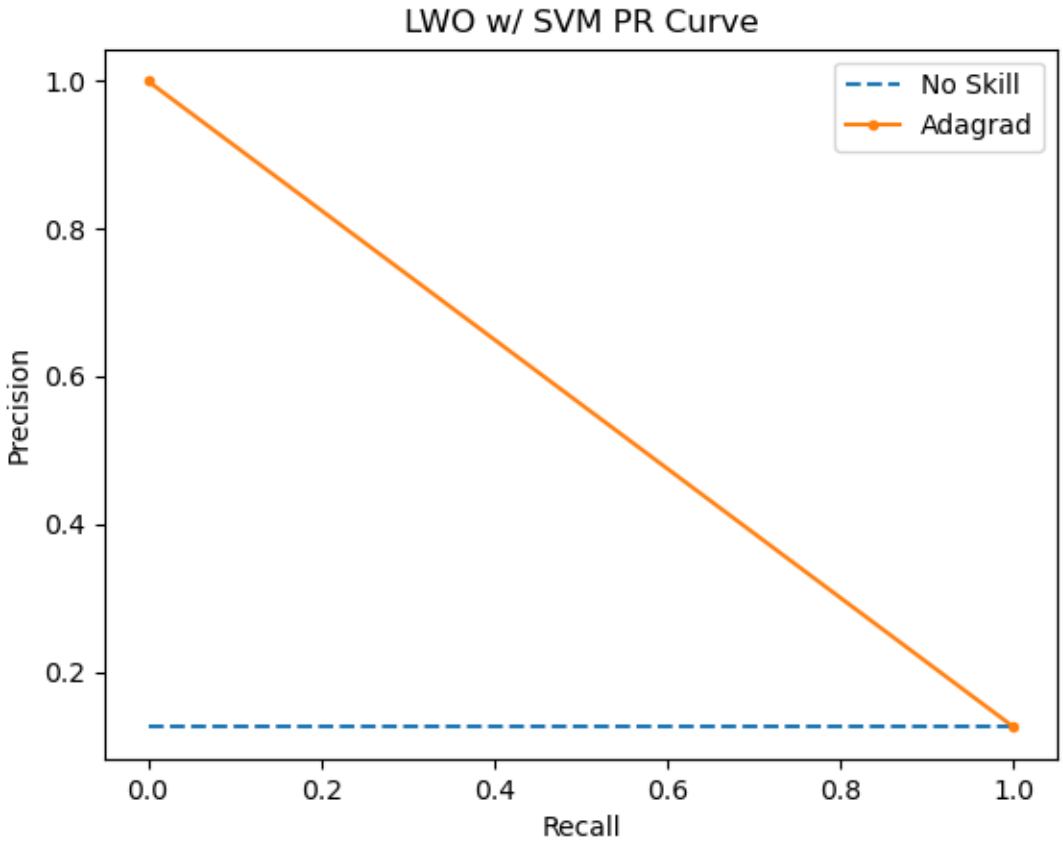

Figure 33: LWO w/ SVM Precision Recall Curve

5.13 -- Deep Neural Network – Optimal Architecture Found

Iris Dataset

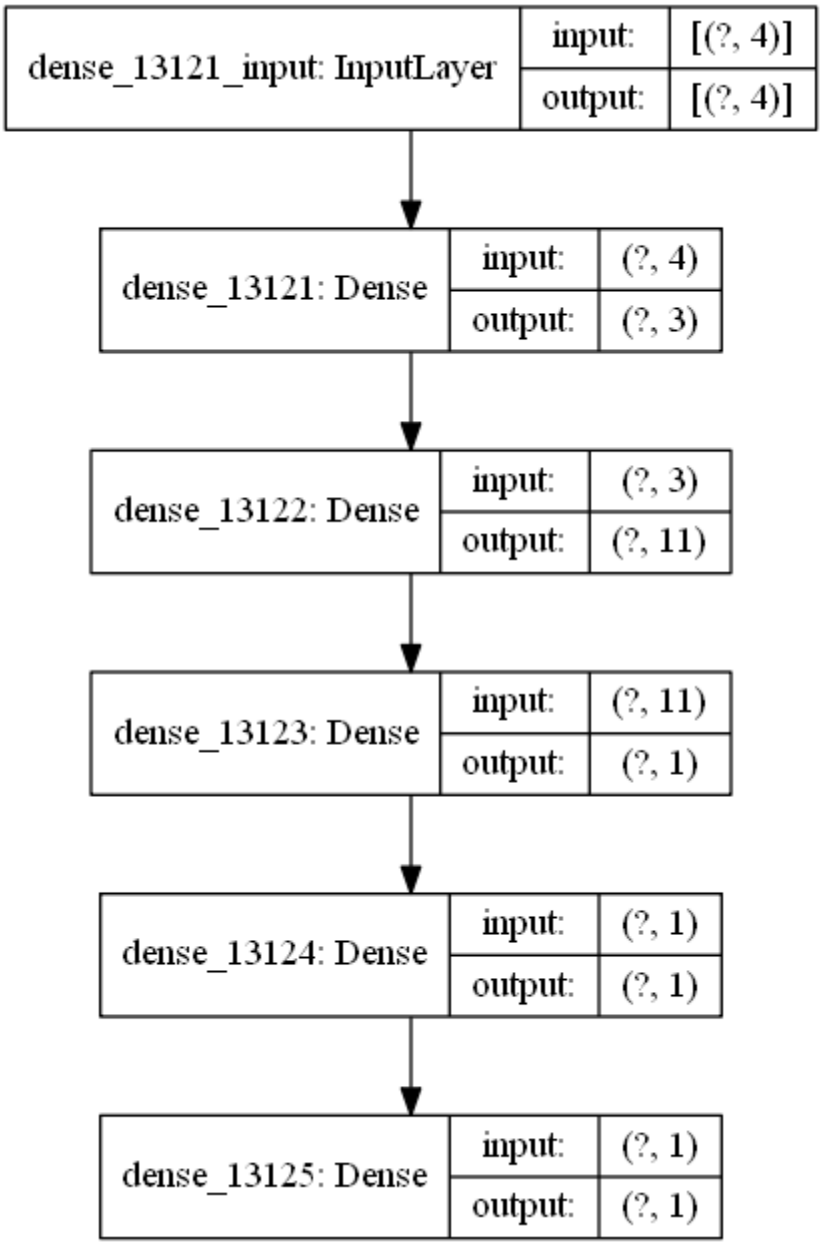

Figure 34: Iris Dataset w/ Deep NN -- Optimal Architecture

Pseudo Survey Dataset

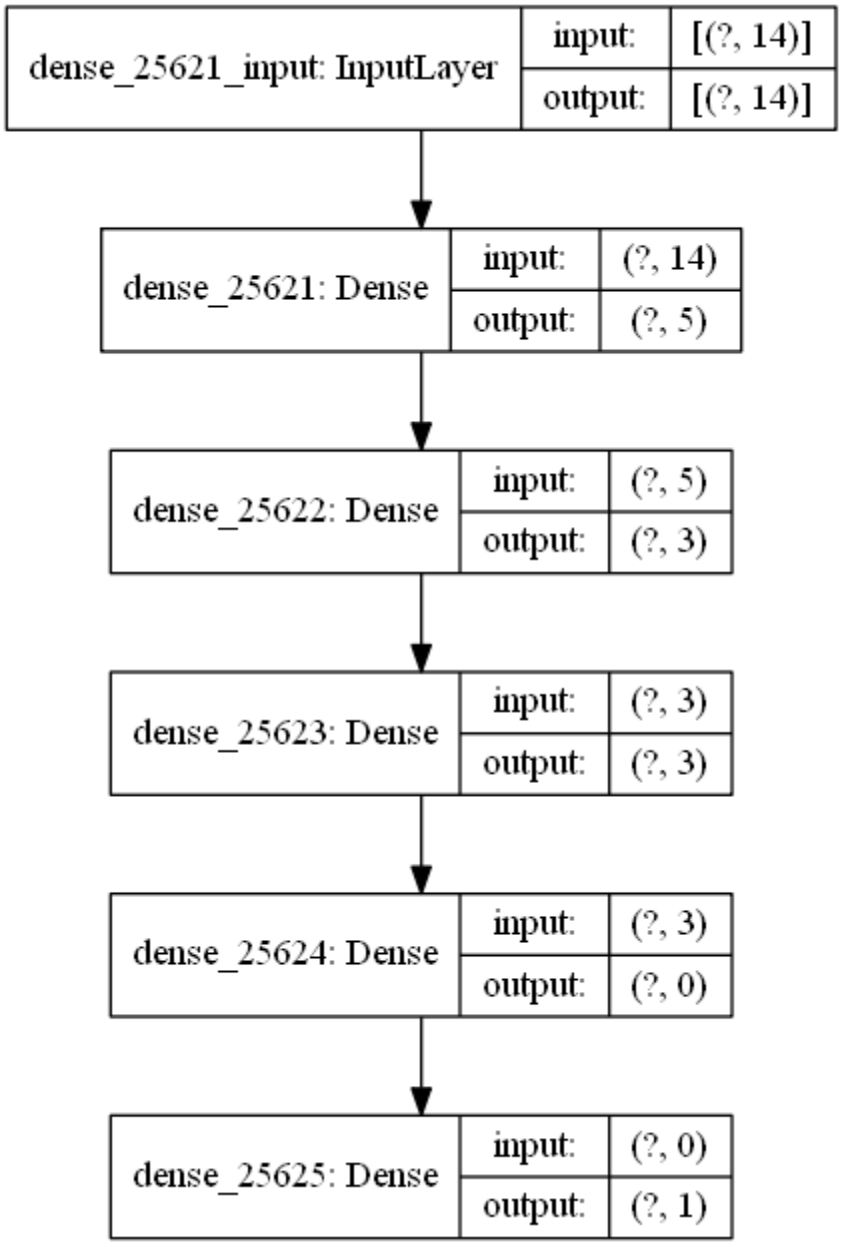

Figure 35: Pseudo Dataset w/ Deep NN -- Optimal Architecture

Real Survey Dataset

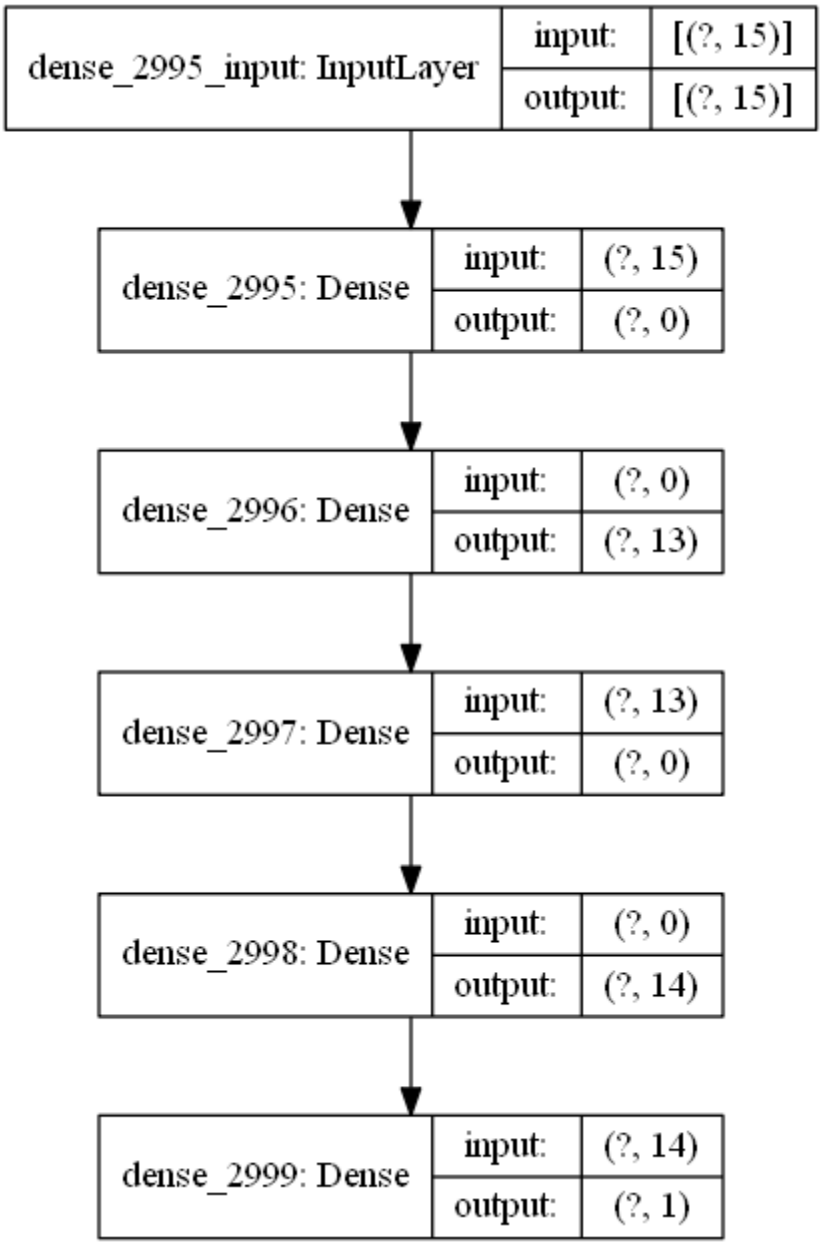

Figure 36: Real Dataset w/ Deep NN -- Optimal Architecture

Comparison

5.14 -- Iris Dataset – Comparison

Table 2: Iris Dataset -- Model Comparison

Machine Learning Method	Training Accuracy	Testing Accuracy
Adagrad (Solo w/ no optimizer) [100 epochs]	31.8%	29.5%
SVM (Solo w/ no optimizer)	N/A	95.56%
Naïve Bayes (Solo w/ no optimizer)	N/A	95.56%
GWO w/ SVM [25 epochs]	100%	100%
GWO w/ Deep NN [10 epochs, 25 iterations]	100%	33.33%
LOA w/ SVM [3 epochs]	N/A	95.56%
LOA w/ Deep NN [10 epochs NN, 3 epochs LOA]	Timeout Error	Timeout Error
Deep Neural Network [150 epochs no optimizer]	37.14%	46.67%
LWO w/ SVM [3 iterations]	62%	42%
LWO w/ Deep NN [10 epochs, 3 iterations LOA]	Timeout Error	Timeout Error

5.15 -- Pseudo Survey Results – Comparison

Due to the nature of potential labels, 'student dropped' or 'student did not drop', the accuracy of randomized guessing is 50% for this dataset.

Table 3: Pseudo Dataset -- Model Comparison

Machine Learning Method	Training Accuracy	Testing Accuracy	F1 Score
Adagrad (Solo w/ no optimizer) [100 epochs]	49.7%	55.9%	0.6262
SVM (Solo w/ no optimizer)	N/A	69%	0.6651
Naïve Bayes (Solo w/ no optimizer)	N/A	65%	0.6789
GWO w/ SVM [25 epochs]	74.15%	77%	0.7140
GWO w/ Deep NN [10 epochs, 25 iterations]	69%	68%	0.5376
LOA w/ SVM [3 iterations]	N/A	100%	0.5505
LOA w/ Deep NN [10 epochs NN, 3 iterations LOA]	Timeout Error	Timeout Error	Timeout Error
Deep Neural Network [150 epochs no optimizer]	100%	87.5%	0.71208
LWO w/ SVM [3 iterations]	72%	64%	0.5505
LWO w/ Deep NN [10 epochs, 3 iterations LOA]	Timeout Error	Timeout Error	Timeout Error

5.16 -- Survey Results – Comparison

Due to the nature of potential labels, 'student dropped' or 'student did not drop', the accuracy of randomized guessing is 50% for this dataset.

Table 4: Real Dataset -- Model Comparison

Machine Learning Method	Training Accuracy	Testing Accuracy	F1 Score
Adagrad (Solo w/ no optimizer) [100 epochs]	1%	26.8%	0.7955
SVM (Solo w/ no optimizer)	N/A	87.5%	0.8167
Naïve Bayes (Solo w/ no optimizer)	N/A	87.5%	0.8167
GWO w/ SVM [25 epochs]	87.5%	87.5%	0.8167
GWO w/ Deep NN [10 epochs, 25 iterations]	95.83%	95.83%	0.8167
LOA w/ Deep NN [10 epochs NN, 3 epochs LOA]	Timeout Error	Timeout Error	Timeout Error
LOA w/ SVM [3 epochs]	N/A	87.5%	0.8167
Deep Neural Network [150 epochs no optimizer]	100%	91.67%	0.8167
LWO w/ SVM [3 iterations]	58.33%	83.33%	0.8167
LWO w/ Deep NN [10 epochs, 3 iterations LOA]	Timeout Error	Timeout Error	Timeout Error

5.18 -- Summary and Conclusions

As the main purpose of this research was to test newly proposed metaheuristic algorithms, it's logical to compare them to more traditional algorithms such as SVM, Naïve Bayes, and Adagrad. For the iris dataset the Grey Wolf Optimizer produced the highest accuracy on testing data, with LOA performing with comparable accuracy as that of it's traditional counterparts. For the pseudo survey data, LOA again performed the best with GWO and LWO performing on par with the conventional machine learning pipelines. Interestingly, the deep neural network performed better than GWO and LWO, a feat not achieved in the Iris dataset. Finally, for the real survey data, GWO performed the best with 95% accuracy, with LOA performing on par with SVM and naïve bayes. LWO performed slightly under these previously mentioned pipelines, but the deep neural network performed its best, above that of Naïve Bayes and SVM, for the least populated dataset.

All three metaheuristic algorithms (GWO, LOA, LWO) proved to be valuable pipelines for the three sparsely populated datasets that they were tested with. Unfortunately, the LWO algorithm came into long training time issues when paired with the deep neural network.

The feasibility of predicting a student drop a course using machine learning pipelines appears to be quite high. Not only did multiple models produce favorable results, including the metaheuristic models; but, the reasonably accurate results were produced from a dataset of only 50 students, 4 of which dropped a course. Assuming more students are able to take the survey over time, it is reasonable to assume the accuracy of each model will increase as well.

As for my prediction of optimizer performance, based on the testing accuracy, the rankings of each optimizer are outlined in the table below. The metric determining overall rank is the average accuracy of the model across the three datasets used.

Table 5: Optimizers Ranked

Model	Iris Dataset Accuracy	Pseudo Dataset Accuracy	Real Dataset Accuracy	Average Accuracy	Rank
GWO (SVM)	100%	34%	87.5%	73.83%	2
GWO (NN)	33%	68%	95.83%	65.61%	3
LOA (SVM)	95.56%	100%	87.5%	94.35%	1
LWO (SVM)	42%	64%	83.33%	63.11%	4

Finally, when operated under the condition that a professor will be capable of inserting their input prior to implementing an intervention tactic, the probability they will be aiding a student in need is high enough to see the merit to augmented intelligence.

Works Cited

[1] R. S. Baker, "Data Mining for Education," Carnegie Mellon University, Pittsburgh, unknown.

[2] R. Lakshmanan, A. Geetha, M. Khalid and P. Swarnaiatha, "Student Performance Prediction Model Based on Lion-Wolf Neural Network," *International Journal of Intelligent Engineering and Systems,* vol. 10, no. 1, p. 9, 2017.

[3] A. Dutt, M. Ismail and T. Herawan, "A Systematic Review on Educational Data Mining," *IEEE Access,* vol. 5, pp. 15991-16005, 2017.

[4] D. Araya, "Augmented intelligence and the future of work," *The Economics,* 2019.

[5] A. Lydia and S. Francis, "Adagrad -- An optimizer for stochastic gradient descent," *International Journal of Information and Computing Science,* vol. 6, no. 5, p. 3, 2019.

[6] S. Ruder, "An overciew of gradient descent optimization algorithms," Insight Centre for Data Analytics, NUI Galway, Dublin, 2017.

[7] Y. Toklu, "An overview of metaheuristic algorithms," *Metaheuristics and Engineering, Bilecik University,* pp. 13-16, 2014.

[8] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software,* vol. 69, p. 16, 2014.

[9] G. Webb, "Naive Bayes," *Encyclopedia of Machine Learning,* vol. 15, pp. 713-714, 2010.

[10] N. William, "What is a support vector machine?," *Nature Biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

[11] B. R. Rajakumar, "Lion algorithm for standard and large scale bilinear system identification: A global optimization based on Lion's social behavior," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, 2014.

[12] M. Yazdani and J. Fariborz, "Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm," *Journal of COmputational Design and Engineering,* vol. 3, no. 1, p. 13, 2016.

[13] S. Hosein, "datacamp," 19 February 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/active-learning?tap_a=5644-dce66f&tap_s=951023-a33697&utm_medium=affiliate&utm_source=kaybajaj&tm_subid1=712877&tm_subid2=91pcnvqaQOKmxdjpfbmcoA. [Accessed 18 October 2020].

[14] A. Dawod, "Active Learning Survey," June 2013. [Online]. Available: https://www.researchgate.net/publication/240918232_Active_Learning_Survey. [Accessed 18 October 2020].

[15] S. Budd, E. Robinson and B. Kainz, "A Survey on Active Learning and Human-in-the-Loop Deep Learning for Medical Image Analysis," Department of Computing, Imperial College, London, UK, 2019.

[16] I. S. Q. A. Sana Fatima, "Analyzing Students' Academic Performance through Education Data Mining. 3C Tecnolgia. Glosas de innovaction aplicadas a la pyme," in *International Multi-Topic Conference on Engineering and Science*, Mauritius, 2019.

[17] R. C. Tanya Joosten, "A Cross-Institutional Study of Instructional Characteristics and Student Outcomes: Are Quality Indicators of Online Courses Able to Predict Student Success?," *Online Learning Journal,* vol. 23, no. 4, p. 25, 2019.

[18] I. J. M. T. Tapani Toivonen, "Augmented intelligence in educational data mining," *Smart Learning Enviornments,* vol. 6, no. 10, p. 25, 2019.