DEEP LEARNING APPROACH AND BIO-INSPIRED COMPUTING FOR SURVIVAL PREDICTION

S.D.V.Prasad
Gitam University
Esteemed Guide And Advisor
sdvprasad554@gmail.com

Tanmay Agarwal
Gitam University
221710305058
tanmayagarwal00010@gmail.com

N.V.Pranay Raj
Gitam University
221710305036
pranay.neethi@gmail.com

R.Sai Anudeep
Gitam University
221710305045
anudeepryali@gmail.com

Jayanth Koganti
Gitam University
221710305018
kgntjynt@gmail.com

ABSTRACT

This research and model fabrication is aimed at achieving a detailed data analysis of underlying deep learning systems of a model, while evaluating the effects of parameters critical to the survival of individuals on board a civilian transportation vessel. Considering the case of Titanic, a British cruise, the analysis will be demystified using systematically formulated and optimised algorithms. Forecasting the survivability rates of diverse groups of populations using a bio-inspired machine learning fabric, to optimise the performance of classifiers is the preliminary objective of this paper. Towards the end, accuracies of various algorithms based on features fed to them will be streamlined and validated against one another.

Keywords : Bio-inspired algorithms, Machine learning, Artificial Neural Networks, Particle Swarm Optimisation

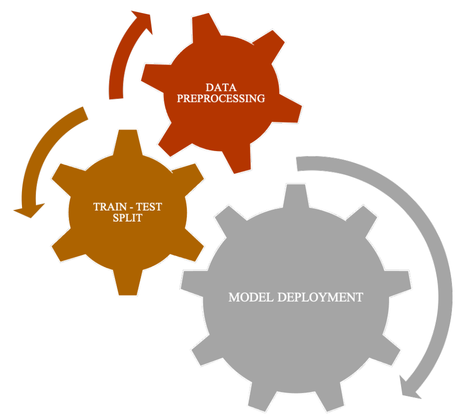

Figure 1: High Level Architecture

1 INTRODUCTION

In the recent industrial era, where science and technology have been growing with an exponential pattern, commercial civilian transportation has turned into a paramount industry. The modes of travel for civilians have never been this elaborate as they are to date. Transportation has become much more standardised and has resulted in greater movement of high density clusters of population. But with this magnanimous scale, there also comes a boon as a result of fiddling with the laws of nature.

Disruptive effects on transportation systems are cause by anthropogenic disasters which impact infrastructures, terminals and modes. Every from of transport in- volves nature as a medium of propagation. Air travel involves movement across the vast blue skies and water transportation involves the abundant water masses on our planet. Whenever mankind has shown pride and pretension on its work, nature has always taught it a lesson.

2 OBJECTIVE

The preliminary objective of our proposed model is to predict the probability of survival of a passenger on board a civilian vessel based on various attributes. Our aim is to minimise the risk in the transportation industry and be impactful to help save lives of people. Based on the projections of our model, disaster management authorities can understand which group of people have a lower chance of survival during a catastrophe, and they can implement safety measures in advance to increase the probability of survival

3 ARCHITECTURE OF PROPOSED MODELS

We propose the use of Custom Artificial Neural Networks and Bio inspired Meta-heuristic Particle Swarm Optimisation Algorithm. Before, moving on to the actual models we first need to walk through the steps of data pre-processing

The broad steps include data manipulation, splitting of the data and testing and fitting of the models. Each of these steps ensure that the consistency in the dataset is maintained and any rogue values are eliminated giving us higher accuracy's during the time of testing and training.

All the steps in the data pre-processing phase ensure that the data is highly consistent throughout. Emphasis is largely made on eliminating any occurring null values. The model does not work well under the influence of null values. Having null values increases the chances of bugs and errors in the data model which is highly non trivial. Also this might be highly non trivial, but null values take up more space than other placeholders

3.1 DATA VISUALISATION

The next phase in this process is data visualisation. Knowing before hand, the attributes which are strongly co-related to each other ensures that the model can be emphasised to use them in the appropriate priority. We can setup a priority in training as to train highly co-related values with much emphasis cause these values have a higher chance of boosting accuracy figures

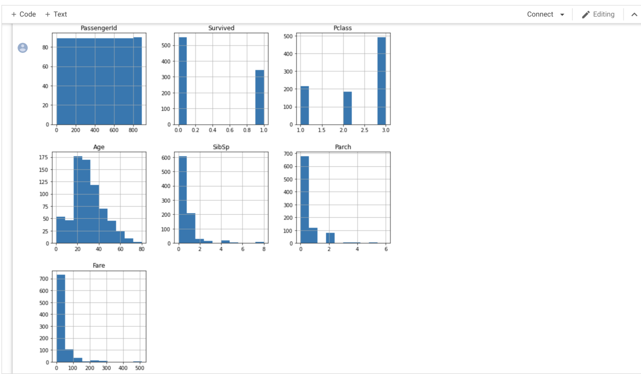

Figure 2: Data Visualisation

Emphasis on highly co-related data yields a higher chance in boosting accuracy metrics as co-related attributes directly impact on the survivability and probability rates.

3.2 DATASET AND TRAIN-TEST SPLIT

We have split the data into 7:3 ratio in which the former part constitutes the training data and the latter the testing data. This ratio is appropriate the provide enough data in the testing phase such that the model neither over fits not does it under fit. This division is a sweet spot for general machine learning algorithms and works like a charm for our proposed as well as base paper implemented algorithms.

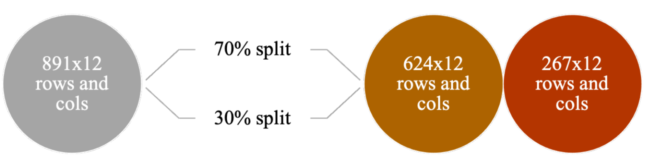

Figure 3: Training And Testing Split

The main dataset contains 891 rows and 12 columns. After performing a 7:3 split the training data consists of the bulk of 624 rows and 12 columns whereas the testing data consists of 267 rows and 12 columns

4 ALGORITHMS

The flagship algorithms that we would like to propose are Custom Artificial Neural Networks and Particle Swarm Optimisation Algorithm. These models have been implemented to their highest working accuracy with respect to the dataset and have been observed to output desired results with tremendous accuracies. In our project we have incorporated industry standards of testing and developing the design of our model

4.1 ARTIFICIAL NEURAL NETWORKS

Initially we will start by implementing our custom neural network. We have implemented a four layer sequential neural network, with the fourth layer being the output layer and the first layer being the input layer.

The first layer comprises of 39 Dense neurons. Dense, in neural networks means that all the nodes in this layer will be connected to each node in the next layer. There is a one to all relation between each node of first layer to every other node in the next layer and vice versa. The next layer is also a dense layer, but with only 27 neurons.

Hence, in total, all the 39 neurons of the first layer are connected to all the 27 neurons in the next layer, which yield a total of 1026 connections between the first two layers

The third layer consists of 19 neurons which are again dense in nature. Which means that these 19 neurons will be connected to each of the previous 27 neurons, amounting to a total of 513 connections between the second and third layer.

Now the final layer consists of only one neuron as it is the output layer. Even this layer is a dense layer which means it is connected to all the previous 19 neurons of the third layer. Hence there are a total of 19 connections between the last and third layer. In total our model has 1558 connections.

Moving on to the activation functions, the first layer uses an activation function called Rectified Linear Unit or simply called RELU. Relu, is preferred over the widely using Sigmoid function because it avoids the problems of vanishing gradients. Relu takes an input and returns zero, if the value is less than zero, or return the input itself if the value is greater than zero.

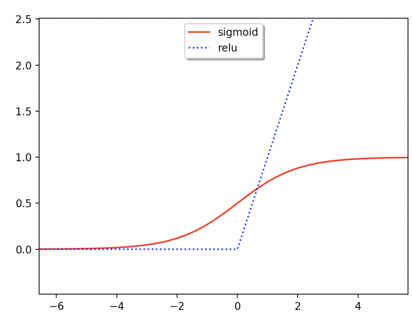

Figure 4: Sigmoid And Relu Activation Functions

The Pseudo Code for ANN is as follows :

- Initialise the first layer with 39 dense neurons and Rectified Linear as the activation function while specifying the input shape

- Initialise the second dense layer with 27 neurons and scaled exponential rectified linear as the activation function
- Initialise the third layer with 19 neurons having softmax activation
- Initialise the final output layer with one neuron having sigmoid activation function

4.2 NUANCES OF ACTIVATION FUNCTIONS

The Rectified Linear Unit activation is as follows :

$$Relu = max(0, x) \quad \text{Where x is the input} \tag{1}$$

Relu, is preferred over the widely using sigmoid function because it avoids the problems of vanishing gradients.

The Scaled Exponential Linear Unit is as follows :

$$Elu = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \tag{2}$$

The Equation for softmax is :

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3}$$

The Equation for Sigmoid is :

$$h_\theta(x) = \frac{1}{1 + e^{-}\theta^T x} \tag{4}$$

4.3 PARTICLE SWARM OPTIMISATION

Particle Swarm Optimisation is a Bio-inspired metaheuristic search optimisation algorithm, that has been developed on the grounds of studying and observing the survival nuances of the wild. It is an iterative computational method which finds an optimal solution in a batch of candidate solutions based upon certain constraints which are enforced as a part of the problem.

Bio-inspired Particle Swarm Optimisation, has been developed and modelled on, after observing groups of animals like a flock of birds or a school of fishes.

Nature has been a source of inspiration for many great endeavours and what better to teach us survival than the nature itself. Survival instinct can be visualised among a group of animals by observing their behaviour and the way animals in the environment communicate with each other in groups and flocks.

One of the main reasons to deploy bio-inspired algorithms is that they are computationally less expensive. When compared to traditional algorithms like the Stochastic Gradient Descent algorithm, which uses first and second order partial differential equations for search optimisation, bio inspired algorithms use linear algebra which is not only less expensive on the system hardware but also faster to compute.

Particle Swarm Algorithm is a metaheuristic (Problem independent), bio inspired search optimsiation algorithm. Observations were made in the mid 19th century that laid grounds to the fact that animals is groups such as a flock of birds, communicate more effectively using natural patterns which increases the likelihood of survival and because of this they are able to survive the wild for substantial longer periods of time. This laid to the foundation of research in Bio-Inspired algorithms.

The greatest strength of particle swarm optimisation is its Parallelizability. This means that the code runs optimally through an

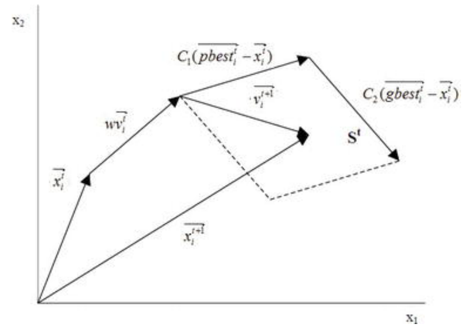

Figure 5: Initial Vector

entire slew of systems, which include single core systems, multi core systems as well as the high performance systems, without any tweaks made to the source code.

Converging towards the technical details of implementation, The main crux of a Particle Swarm Algorithm include Position Vector, Momentum Vector, Fitness Function, Particles best position, Global Best position. The position vector is the goal of the optimisation problem, The fitness function is the objective function which determines the position, the velocity vector represents the speed at which the swarm moves.

The Pseudo Code for Particle Swarm Optimisation is as follows :

- Initialise a random population swarm with random volume of particles in each swarm
- Each Particle represents a potential solution
- Initialise the particles with random velocities and position vectors which are the global and local best vectors
- Initialise the fitness function
- Update each generation of the particles in accordance with the optimality of the fitness function.

In an abstract view to optimise a problem tailored for particle swarm optimisation, we initialise a solution space with random population of points, which is analogous to our input data. the next step is updating the position and momentum vectors of each point, enforcing the above updation relations, and generating new solution space which is more optimal than the previous.

For each particle in a population, we randomly initialise the position and momentum vectors. Based on these values a fitness function is calculated which describes the optimality of our problem. Then the vectors are updated based on the above rules and then the local fitness function will be calculated. If this new fitness function is more optimal than the previous function, the new local function will be set as the initial function in the next generation. With this small local optimisations translate to large global optimisations

4.4 UPDATION RULES IN PARTICLE SWARM

The Particle Swarm updation rules are as follows

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \tag{5}$$

$$V_{ij}^{t+1} = wV_{ij}^t + c_1 r_1^t (pbest_{ij} - X^t ij) + c_2 r_2^t (gbest_j - X_{ij}^t)$$

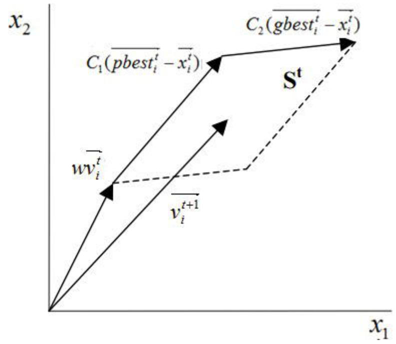

Figure 6: Updated Vector

The position best and global best vectors are checked in accordance to the fitness function which consists of optimisation rules of maximisation and minimisation

Each updated solution space is termed as a generation. We iteratively update the position and global and local solutions of every generations, while keeping every future generation more optimal than its ancestors. In each generation more and more points, which are termed as particles, get more and more optimal and the small local optimisations, turn into vast global optimisations over the course of iteration.

5 RESULTS AND VALIDATION

Having implemented all the algorithms successfully, we have to now analyse all the accuracies which we have obtained by training and testing our model. The below tabular illustrates an overall accuracy picture which will give us detailed information for analysis purpose.

Table 1: Accuracy Validation (In Percentile)

Algorithm	Accuracy
Artificial Neural Networks	86
Particle Swarm Optimisation	83.4
Support Vector Machines	82.3
Logistic Regression	82

From this we conclude that our implemented **Custom Artificial Neural Network algorithm gives the highest accuracy of 86 percentile** while giving the lowest amount of False values. The accuracies vary with a certain degree on margin of error because of system constraints and definition of functions. This margin of error is minimal as it was intended.

Both our flagship models performed highly accurately while giving satisfying results. These results may vary after changing certain hyperparameters and depends upon the type of system used for testing as well.

Our other main reasons to deploy bio inspired algorithms is that they are computationally less expensive. When compared to traditional algorithms like the Stochastic Gradient Descent algorithm, which uses first and second order partial differential equations for search optimisation, bio inspired algorithms use linear algebra which is not only less expensive on the system, hardware but also faster to compute.

We have successfully used the biggest strength of particle swarm optimisation which is parallelizability. The same algorithmic definition can be run with minute to null changes on various different classes of machines like high performing computers or single and multi-core computers. This provides a degree of freedom that no other algorithm does.

6 CONCLUSION

Nature has vividly captured the zeitgeist of survival and divulgence, throughout generations and has bestowed mankind with the wealth of its knowledge. It has been a driver for sourcing inspiration into the development and deployment of highly optimised algorithms which happen to solve the problem present due to classical algorithms

We have indigenously manifested the problem of optimising disaster management predictions in the case of natural or artificial catastrophes on board a civilian transportation service vessel.

We conclude that our models accurately predict the survivability of a passenger based on certain metrics provided to our custom learning algorithms, which can be used to provide a certain degree of refuge and increases the likelihood of survival of each individual

Based on the projections of our model, disaster management authorities can understand which group of people have a lower chance of survival during a catastrophe, and they can implement safety measures in advance to increase the probability of survival

The formulation of this problem has been pertaining to a degree of uncertain variance and mean covariance with nominal risk exposure enforcing semi infinite constraints on partly infinite degree of the problem range

With all the perils that has been going on in the world, during the pandemic which has been set in, which has a death rate reaching unconscionable highs, protecting human lives and safeguarding interests has never been to this paramount degree. While we intricately formulate our models, we cannot be oblivious to the fact about what's going on outside the door of our rooms, in the outside world, were people are suffocating and are being denied a proper health care. Never has the world in recent times be in such a state of helplessness

7 CITATIONS

The dataset has been procured from Kaggle [5] with reference to our base paper which is written by Karman singh [6]. We have studied various works which have been executed on this dataset such as works done by Ekinei [1] and Erik Lahm [9]. We have also studied some of the brilliant work done in the field of hyperparamenter optimisations such as Singh [8] and Han [3]. Algorithm fine tuning demonstrated by the works of Peng [7] and Xiao [10] were analysed in their fine print. Spectral algorithm work by Zhao and data mining techniques by Jain [4] were also referenced. The work done by Farang [2] on the datset was highly influential.

ACKNOWLEDGMENTS

This work has been stimulated by **Gitam University** Hyderabad, performed under the esteemed guidance of **Mr.S.D.V.Prasad** in the department of Computer Science and Engineering during the month of May, 2021

REFERENCES

[1] E.Omurca Ekinci and N.Acu. 2018. A comparative study on machine learning technqiues using titanic disaster dataset. *7th International Conference on Advanced Technologies* (2018).

[2] Nadine Farag and Ghada Hassan et al. May, 2018. Predicting the survival of titanic kaggle disaster dataset. *ICSIE'18, Procedings of the 7th International Conference Software and Information Engineering* (May, 2018).

[3] Claudio Han, Jun Morag. 2007. The Influence of sigmoid function parameters on back propogation learning. *Natural to Artificial Neural Computation, Lectures on computer science* (2007).

[4] Nikita Jain and Vishal et al. 2007. Data Mining Techniques a Survey Paper. *International Journal Of Research and Technology* (2007).

[5] Kaggle.com. 2020. Titanic Disaster Dataset. *www.kaggle.com* (2020).

[6] Renuka Nagpal Karman singh and Rajni Sehgal. 2020. Explanatory data analysis and machine learning on titanic disaster dataset. *IEEE Explorer* (2020).

[7] Juhua Chen Peng, Wei and Haiping Zhou. 2017. An implementation of ID-3 Tree Decision Learning algorithms. *From web.arch.usyd.edu.au/wpeng/DecisionTree2* (2017).

[8] Shipra Saraswat Singh, Aakriti and Neetu Faujdar. IEEE, 2017. Analysing Titanic disaster using machine learning algorithms. *2017 International Conference on Computing, Communication and Automation (ICCCA)* (IEEE, 2017).

[9] Eric Lam Stanford University. May, 2020. Predicting the survival of titanic kaggle disaster dataset. *Disaster CS229, Titanic Machine Learning* (May, 2020).

[10] Ying chao Xiao. 2014. Two Methods for selecting Gaussian Kernel Parameters for one class SVM and their applications to fault detection. *Knowledge Based System 59* (2014).