 doi:10.20944/preprints202107.0016.v1

Comparison Between Tailor-Made-ANN Techniques and [1] Fuzzy c-Mean Clustering Technique in Industrial Laborers' Accident-Rates Prediction Modelling Based on Human Factors

Muhammad M.A.S. Mahmoud – Independent Researcher

Abstract This paper attempts to compare two different approaches to solve the problem of accident rates prediction based on human factors for industrial workers. One of the methods has already been done using Fuzzy c-Means Clustering and proved to be working with decent results. The second method which will be covered in this paper is using Artificial Neural Networks. The primary goal of this work is to insure that ANN will work efficiently in such prediction problem. The second goal is to reveal the fact that which one of the two selected methodologies is better at defining the estimation of accident rates among people who work in different industrial fields. The purpose has been achieved when the outcome of the ANN was obtained and compared accordingly with the output of the research previously carried out with Fuzzy c-means clustering method. Comparing the outcomes of these two different methods gave an immense insight on which features are more important than others when it comes to laborers properties with completely different background such as varying levels of health, knowledge, experience, training and physical properties. At the end of the research, it becomes clear that accident rates estimation for laborers with properly trained Artificial Neural Network gives better results when it is compared with Fuzzy c-Means Clustering method. Standard deviation method was used to calculate the validity of ANN technique. The result was compared with Fuzzy c-mean clustering technique. Impressive improvement of 8.8% in the accident rate prediction was achieved using Tailored-Made-ANN.

***Index Terms*— Industrial Laborers, Accident rate, Artificial neural network, Human factor, Predictive models.**

I. INTRODUCTION

Nowaday an increasing number of industrial work areas creates more job opportunities than it has ever done. Many people occupy these kinds of labor-oriented jobs in which there is a very high chance of getting hurt or damaged. This also can be associated with lethal injuries or disabilities which will lead to the end of work of many people. It is not only undesirable for individuals but also for their companies as their reputation is also on the line. These types of accidents happen every day in large-scale industrial projects. Despite all the precautions which are usually taken by the authorities it is still impossible to predict and prevent not only the accidents occurrence but also the severity level and its cost implications (Evanoff B, 2002) (Handley W, 1977) (Ashok Kumar Bansal, 2016) (Choi, S.D. ,2006) (T.A Yusuf,2015).

The main problem here is the inability of forecasting the details of accidents beforehand. Information such as date, location, which laborers might possibly be damaged, which characteristics of a person plays a more important role in an accident, financial cost of an accident, etc. is what we need to have for evaluating the situation and making decisions. Unfortunately, the majority of these data will not be available any time before the accident happens. However, it is possible to build a system that can predict and approximately estimate the rate of accidents which can happen due to human factor and gross errors. In addition, the data available on construction site accidents are neither accurate nor complete, due to the absence of a reliable accident reporting and recording system.

The majority of the work accident happens due to human factors according to several kinds of research and case studies. For example, human ignorance and arrogance are well-known drivers of job accidents. If these human factors, and other effective factors, are taken into consideration, a mathematical model can be obtained for the labors to represent relation between the most effective properties in accidents and the expected rate of accidents. (Strong, 1987) (Wilson, H.A, 1989) (Mohamed MAS Mahmoud, 2012) (Jesús Domech Moré, 2007) (Hidetake Sakuma, 2002) (Douglas A. Wiegmann, 2001) (Douglas A. Wiegmann, 2005) (Kay Yong,2007)

There is only a few number of researches that are done to find a solution and build a system to predict accident rates in industry and they are offering an incredibly little amount of resources to work on. The industry really is in the big need of such work to optimize their safety standards and help to utilize the resource as best as they can. Talking about the gap in the literature, it is also helpful to consider creating an electronic platform to make the resources and the data, that are used in such researches all over the world, available for those who desire to make different comparisons with different techniques wishing to find a better solution.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

The importance of such researches is not only brings more safety to workers, but also it guides us to enlighten our understanding of regulations, rules and break our misconceptions and myths about safety [8], (Mohamed MAS Mahmoud, 2012) (Eunsuk ChoiEunsuk, 2019) (A. J.-P. Tixier, 2016) (Youhee Choi, 2018) (Sobhan SarkarSobhan, 2016) (P. Hämäläinen, 2006).

As previously mentioned, the obvious solution is to build a mathematical model which can predict accident rates accordingly with previously supplied data. It might sound easy, however, taking into consideration that many have tried to accomplish but they got few concrete results, which raises the question that how possible is to generate such a model and what are the methodologies that are need to be implemented to improve the existing models to achieve better results. (A. Lukacova, 2014) (F. Pereira, 2013).

Fuzzy (multi)measures are used in many problems such as decision making (Grabish, 1995), accident rate in the predictions of osteoporotic fractures (Pham, 2008) or image processing (Costarelli, 2020). In (Mohammed MAS Mahmoud, 2012), fuzzy measures are used to build a human model in accident rate estimation in construction market by a fuzzy clustering technique.

In this paper, we selected one research from the few published researches that has tried to solve the problem of accident rate prediction for laborers using a model based on fuzzy clustering c-mean technique. That paper provided reasonable validity for the results. However, the achieved results were satisfactory as initial stage only in that research. But, still there is room for results improvement. In this paper, we will try to obtain better results but by using Artificial Neural network. (Manar M. Sabry, 2002)

Artificial neural network technique was used in many researches to analyze and predict the accidents related to traffic. However, the problem of accident rates estimation that is specifically related to laborers, was never been tried to be solve with ANN method, which raises the question of how efficient it will be for such a different type of problems. Looking at the previous experiences of problems that solved using ANN, this technique is good in case adapt the model to specific training data is required. However, one may still doubt that it is the proper method for problems with high nonlinearity. The new ANN model might not give the result 100%, but the useful thing is that it can be retrained with techniques like reinforcement learning to upgrade the model to a new level [24-30]. (A. P. Akgüngör, 2009) (Mohd Zakwan Bin Ramli, 2001) (Abbas, M., 2010) (F. Rezaie, 2011)(Borja García, 2018) (Khair S. Jadaan, 2014).

In Section II, brief review for Accident Rates Estimation Modelling Based on Human Factors Using Fuzzy c-Means Clustering Algorithm is provided to obtain the input data, and hence the feature matrix, that required for the proposed ANN model. In Section III, prototype model for ANN is used to estimate the accident rate and to evaluate the satisfaction of the results. Then, Section IV explains in steps the structure for ANN modules to create framework. In Section V, ANN Tailor-made framework is used to estimate the accident rates. The results and the conclusion are given in Section VI and VII respectively

II. Data Collection and Feature Matrix

One of the most challenging tasks of doing scientific research is to obtain solid, consistent and verified data. It can easily be said that the primary resource of the paper is the data.

It is required deep research and intensive analysis to find out which features play the most important role in accident rates. According to the studies made previously on the topic, it has been found a reference table which contains the most influential factors that should be taken into consideration when designing a model for accident rate estimation for labors (Manar M. Sabry, 2002).

In Table I, main features are illustrated. As it is clearly shown, the data collection process focused on the laborer, himself, and his accident rates during his years of work experience as an expert source of data. The way that the data collected, as mentioned in the same reference, was by survey method constructed based on these main features. A reputable industrial construction company has accepted to carry out such a survey among the workers so that they can get more detailed information about safety preparation level. It is also worth to mention that the survey had been carried out anonymously.

The data collected was used to construct the Feature Matrix (FM) which is one of the most important parts of the study since we can convert human factors and properties to digital variables using this matrix. By this way, we can easily deal with the data to be used in any arterial neural network model. The columns in the FM matrix represent the property variables that were obtain from the survey. The rows in matrix reflect the different features of the employees who were chosen for the survey interview. Each linguistic feature in the matrix was presented by 1-5 scale to convert the linguistic meanings of the feature used in the survey into respective number presenting the weight of each worker-feature. Table II illustrates sample of feature the matrix.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

TABLE I
MAIN FEATURES AFFECTING THE LABORERS' ACCIDENT RATE

Feature	Description	Feature	Description
F2	Weight/Height	F13	Need for Safety Gear
F3	Optical status	F14	Indoor Work
F4	Hearing Ability	F15	Office Work
F5	General Health	F16	Outdoor Work
F6	Adherence to Safety	F17	Level of Boredom
F7	Education	F18	Salary on time
F8	Overtime work	F19	Level of Training
F9	Mental work	F20	Level of Safety
F10	Manual work	F21	Noise level
F11	Work type	F22	live with family
F12	Hazard level	F23	Communication language
FW1 Accident rate/ labor experience years			

TABLE II
SAMPLE FEATURE MATRIX

Feature / Case No.	F1	F2	F3	F4	F5	F6	F7	F8
S1	1/12	71/160	6/18	5	5	4	5	2
S2	5/12	77/170	6/60	5	4	3	5	3
S3	1/21	90/175	6/6	4	5	5	5	5
S4	0/8	54/165	6/60	4	5	2	5	20
S5	6/3.5	68/187	6/36	3	4	3	5	0
S6	10/11	85/177	6/6	4	4	3	5	10
S7	2/19	76/173	6/60	4	5	5	5	14
S8	18/25	72/170	6/18	3	4	4	4	4
S9	45/14	80/176	6/6	4	4	3	4	8
S10	3/20	81/174	6/6	4	4	5	4	1

III. USING ANN ROTOTYPING MODELLING FOR ACCIDENT RATES ESTIMATION

Initially, an investigation was necessary to find out whether it is possible to predict accident rate using artificial neural networks, and wither will be opportunity to improve the results?

To check the possibility of doing such a prediction in ANN, a prototype version of ANN by using readymade machine learning libraryKeras with TensorFlow is used. The initial code is written in Python shown in (Fig. 1) PEYTON, R. X, 2019)

```
train_df = pd.read_csv("main_data.csv")
train_df.head()
train_X = train_df.drop(columns=['FW1'])
train_X.head()
train_y = train_df[['FW1']]
train_y.head()
model = Sequential()
n_cols = train_X.shape[1]
model.add(Dense(20, activation='relu', input_shape=(n_cols,)))
model.add(Dense(20, activation='relu'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')

early_stopping_monitor = EarlyStopping(patience=5)
model.fit(train_X, train_y, validation_split=0.1, epochs=50,
callbacks=[early_stopping_monitor])

test_X = pd.read_csv("test.csv")
test_y_predictions = model.predict(test_X)
print(test_y_predictions)
```

Fig. I. Prototype Python Code for ANN model

The algorithm starts with reading the input data from the spreadsheet file. Header columns are removed and input and output separated from each other using "drop()" function. Model is created using a single class structure called "Sequential()". Then layers with a number of neurons and activation functions are added to the ANN model. As a learning method, ADAM and cost function mean squared error is selected. Keras framework also provides the training stage with a neat function called "EarlyStopping" which is used for delaying the finish of the training phase and causing much better convergence in the neural network model.

After all the training testing, all it takes to test the model and predict real results is just one line of code which is done using "predict()" function.

Prototype neural network model gave fair results from the first time. However, there was room for improvement showing that better results for the estimation of accident rates can be obtained by using ANN.

In this stage using prototyping tools are end, it was decided to start implementing the similar steps using the custom-made neural network framework.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

IV. Design of Artificial Neural Networks framework

In order to build a neural network model based on some dataset, firstly, we need a tool to create the structure of it. Taking into consideration that the tool should also allow visualizing the neural network so that we can troubleshoot any kind of problem while updating its parts. It is a hard task to create fully functioning neural network with properly working features.

To make this chapter more understandable, simple task will be assigned to try to solve it using the neural network framework as we build it. The task is solving four input XOR. The input will be four digits of which value can only be either one or zero, and as output, the neural network should return either one or zero.

Building the custom artificial neural network builder and visualizer tool was not enough on its own. The process of calculation, training, and testing requires a live analysis of the data processed in the network. There is no direct way of adjusting different variables like the learning rate, number of hidden layers, and the number of neurons in each layer. Therefore, we also need a graph tool to visualize the figures with time or sample cases. Hence, the creation of a simple graphical interface tool is necessary (Amirasln Bakhshil Muhammad M.A.S. Mahmoud,2019)

Following are the steps and structure of modules to create this enormous framework:

A) Base neural network structure
B) Machine learning in ANN
C) Dataset format
D) Neural network visualizer
E) Custom graph tool
F) User interface framework
G) End-user implementation in code
H) Result

A. Base Neural Network Structure

The foundation of the framework required some early investigation of similar tools available on the market. The research showed that the tools currently available for similar applications are divided into two groups in term of their internal structure. The first group takes a shortcut approach and uses linear algebra methods to resolve the neural network calculations. Creating multidimensional matrixes and multiplying them takes very little time. This is a great advantage. On the other hand, it has one major drawback that this method keeps the neural network operations as a black box. It means we cannot understand thousands of the gibberish figures inside a grand matrix. This is not useful to build the framework. It also brings some challenges when visualizing the state of the neural network as we cannot directly identify which figure stands for which weight. Hence, as the designer of the framework, the decision was taken to create framework using the second option which is the object-oriented approach. In this approach, instead of multiplying matrices using linear algebra, the matrices calculations are solved using programming tools called loops. It is a little bit of time-consuming operation though. In return, we get readable and understandable data and figures stored in the logical data structure.

Each data structure is named properly in the code which is also shown in the UML diagram below so that even an individual who is not aware of programming principles can understand which class is responsible for which actions (Fig. II)

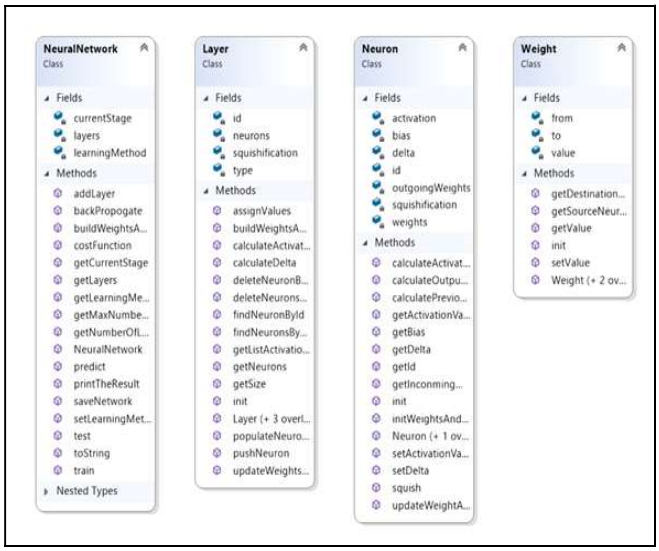

Fig. II. UML Diagram of Base ANN

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

As it is clearly seen from the diagram, the core class for the ANN is Neural Network class. It contains layers and all the tools for accessing them. It is also full of methods which will help the end user to apply certain learning methods, cost calculation functions, and testing methods. Layer, as its name suggests, is just a virtual container for neurons and it also contains all the necessary functions to satisfy the requirement of encapsulation.

Neuron class is the second most important class in the whole framework. It does implement lots of functionalities including the calculation of activation function, adding bias and multiplying weights. It also stores a pair of vectors of weight pointers. One is for incoming weights and the other one is for outgoing ones; and finally, Weight class is the simplest yet the most influential class in the project. What all the neural network is that is a bunch of weights and biases collected in a data container. Any neural network model can sufficiently be recreated with this data. Objects created from Weight class contains three important properties. From which class it is coming. To which class it is going, and what is the value it is holding.

B. Machine Learning in ANN

After introducing the basic structure behind artificial neural networks, we can now discuss how the process of training can be realized using programming principles. But first some decisions must be taken such as; which learning method and cost function will be applied, what type of activation functions should be available.

After making the decisions, it was the time to build the classes one by one. Starting with the activation function, I created an abstract class name "Squishification" class, which is the ancestry base class for all activation functions. I gave it this name because its primary aim is to map a wide range of numbers into very narrow and squished interval.

In the framework, two different activation function instances have been implemented. The first one is the "Sigmoid" activation function and the second one is "ReLU" activation function. Although they called function, however they are created as a class in the framework. The inheritance relationship can clearly be seen from Fig. III illustrated.

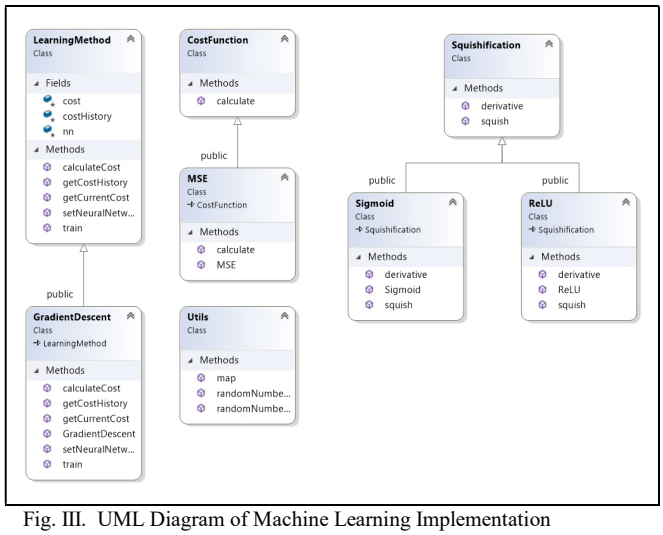

Fig. III. UML Diagram of Machine Learning Implementation

Each activation function class derivative contains three main functions. One of them is the primary constructor of the class. The other two are the "squish()" and "derivative()" functions which are the virtual abstract methods from the "Squishification" class.

"CostFunction" plays a crucial role during the convergence period of training. For using one of the principles of object-oriented programming called polymorphism, "CostFunction" class was created as an abstract one and derived more specific cost function which is "Mean Squared Error" method (MSE).

The most important driver of the machine learning in artificial neural networks is the learning algorithm. There are many types of learning algorithms, but the most suitable one for the training was selected. It is "GradientDescen" method which uses "backpropagation" to redistribute the expected values to each neuron and updating the weights and biases accordingly. The amount of change applied to each weight and bias is calculated reference to the value obtained from the cost function set for the learning method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

C. *Dataset Format*

As we build our own custom ANN tool, we also need to create a custom dataset standard and a tool for reading data from a given resource and apply it into the ANN automatically. To implement that two classes were created called "DataSet" and "Sample". The internal structure, fields, and methods can be seen in Fig. IV.

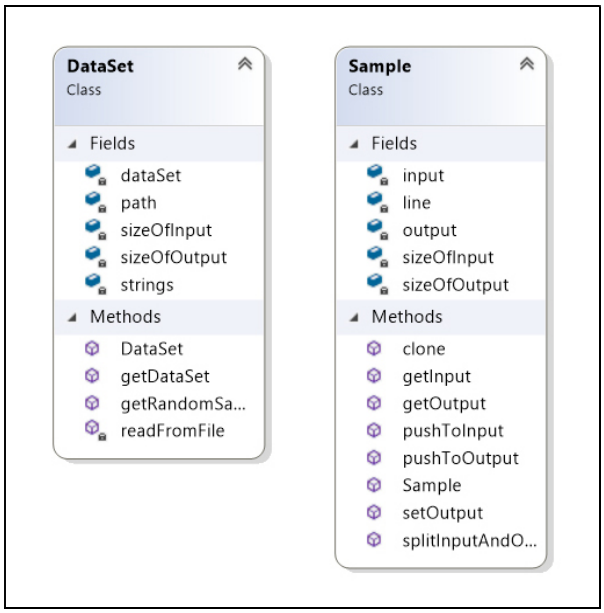

Fig. IV. UML Diagram of Dataset Format Implementation

As we declared in the introduction, we have set a task to solve with this framework. So, if we take look at the dataset prepared for this XOR solving network (Fig. V), we can see several rows. Some of the rows are hidden intentionally. Each row represents one case for the neural network and according to our condition, we have four input values and one expected output values for each case. So, the number of digits in each case is five meaning the first four is for input values and the rest is for the output value.

```
0 0 0 0 0
0 0 0 1 1
  …
  …
1 1 1 0 1
1 1 1 1 0
```

Fig. V. Dataset Format Example

D. *Neural Network Visualizer*

Up to this point, the most important functional parts of our framework; properly data taking, training, testing and predicting artificial neural network is ready. Now we will focus on the design of the visualizing module.

As it can be seen from Fig.VI illustrated below, all the elements from the base structure of the neural network have been duplicated for their visual version. These duplicates are only for representing a visual image of real neurons, layers, and weights. Properties and methods are all about the visual features of the neural network such as the diameter of a neuron, colors, gaps between horizontally and vertically spaced weights and neurons, margins, paddings, texts, text colors, etc. During the initialization process, one of the important steps is to map the mathematical state of the artificial neural network to the visual version of it. To implement this kind of logic, it needs to start with a base class called “VisualObject” and it is an abstract class. Then create the other classes namely “VisualNeruon” for “Neuron”, “VisualWeight” for “Weight” and “VisualLayer” for “Layer” class. The visual versions of those classes are all derived from the “VisualObject” class as it can clearly be seen from Fig. VI shown below. It is mandatory to implement the “draw()” abstract virtual method. Without it, there would be no way to draw the object in every cycle of the loop.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

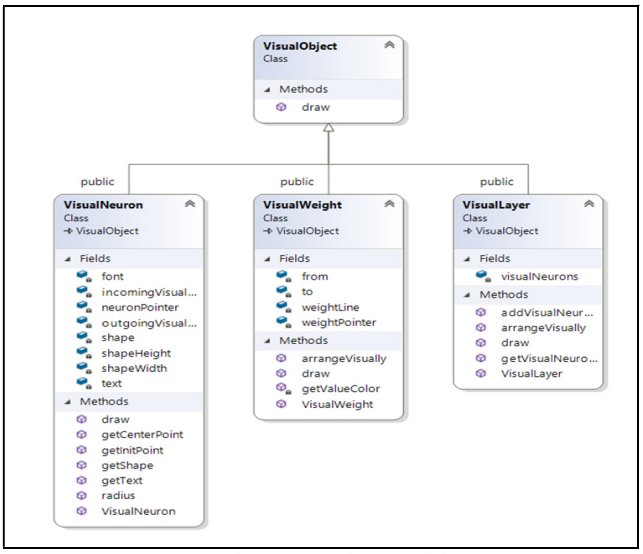

Fig. VI. UML Diagram of ANN Visualizer Module

E. Custom Graph Tool

After creating the tool for representing the state of the artificial neural network visually, another visual interface for representing the cost function of the neural network need to be built. As it was mentioned in the beginning, there is no way of evaluating the progress and optimizing the variables without properly monitoring the cost values of the network. The best way to visualize a vector depending on case number is to use a graph tool. Therefore, as a part of the framework, custom graph tool is needed to be built. It does not include many features as other graph interfaces do but it does the job with consuming as minimum memory as possible while fitting the framework much more smoothly.

Fig. VII illustrates the graph module with main driver class named "Graph" class. The axis of the graph is represented with the base class called Axis class and it is an abstract one to derive both of "X_Axis" and "Y_Axis" from it. In that case, it is not required to replicate the same code throughout two classes.

Since the numbers attached to the X and Y axis will dynamically be updated, A new class called "AxisNumber" was generated to monitor the value of the numbers on the axis. This new class also helps to animate the values in the graph much more easily.

Another problem was to have a data container for the graph. It is better to have an isolated data container so the module can be mobilized and abstract.

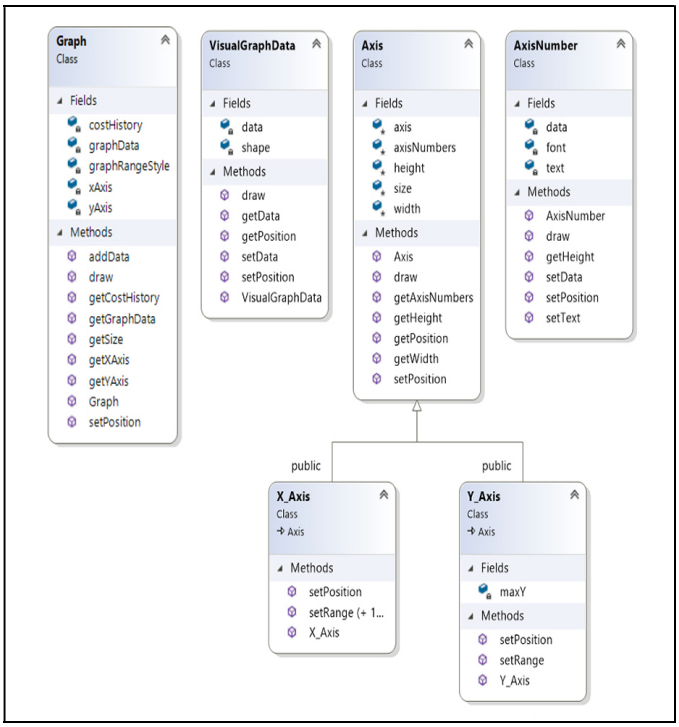

Fig. VII. UML Diagram of Custom Graph Tool

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

The only thing left was to decide what logic we want from our graph to work. There were two options. The first one was to build it in such a way that old values that have already shown on the graph would be pushed into left, then new values would also come from the right side of the graph. This functionality was implemented in framework "LastNRange" class as shown in Fig. VIII. The second method was to keep all the values in the visual sight and stack them on top of each other horizontally, from left to right; and this functionality was implemented on "FullRange class".

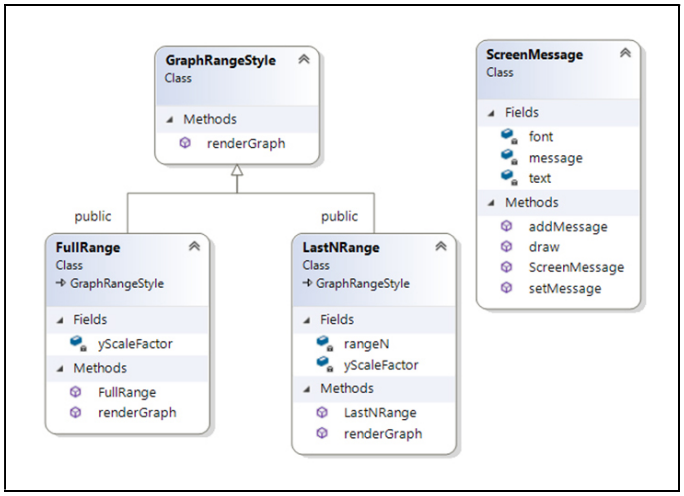

Fig. VIII. UML Diagram of Graph Style Implementation

Both of these functionality classes are derived from "GraphRangeStyle" base class which in itself have"renderGraph()" abstract method.

Another important point was to create a tool to tell the current value and stage of the neural network working completely independent of the graph that is working in the background. To create this type of feature, “ScreenMessage" class is need to be created which was a text message container stacked on the left top corner of the screen demonstrating the live values of the cost function and representing the stage of the progress of the neural network. The stages that can be shown in the screen message are the following:

- Created
- Initialized
- Training started
- Training…
- Training ended
- Testing started
- Testing…
- Testing ended

Fig. IX illustrates the last N range style graph representation. Notice the range shown on the X axis as it starts from 4150 and goes up to 4470. As the progress continues to improve the number are updated keeping the same range interval. This type of graph is especially useful when working with huge datasets and the importance of the very old and initial values are ignorable.

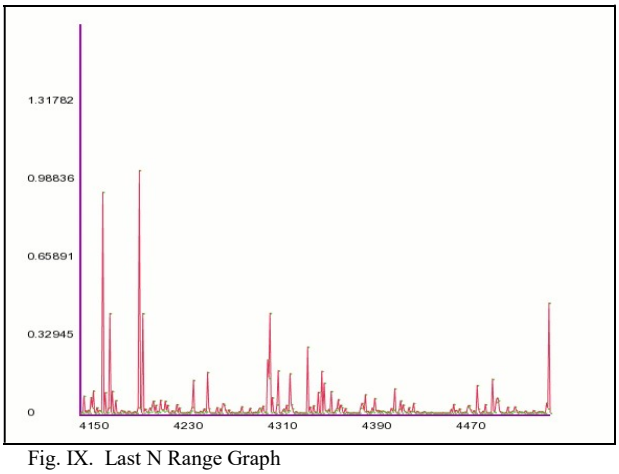

Fig. IX. Last N Range Graph

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

Fig. X demonstrates the second type of graph which is full range type. As it can see from the origin point, the x-axis starts from the value of 0. This means that even the old values are included in the result. This style is preferred when the overall performance of the network is needed to be measured. Usually, for detecting the slope of the convergence we use full range graph. However, the major drawback of this type of graph style is that it takes too much space in memory due to the storage of all the numbers at once and adds some delay to the visual renderer as it increases the number of dots to be drawn on the screen.

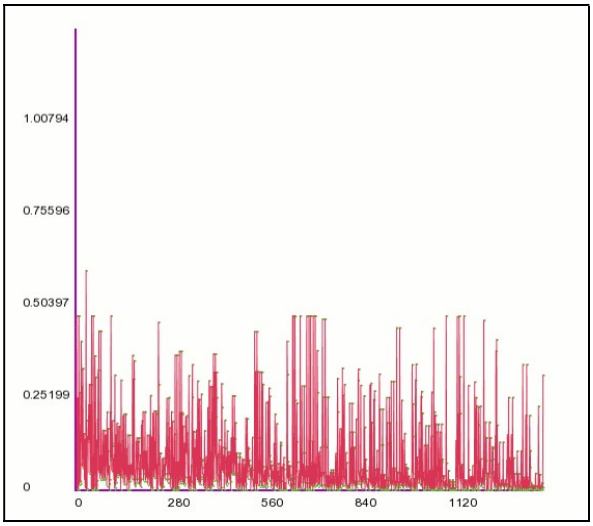

Fig. X. Full Range Graph

F. User Interface Framework

After completing the graph visualizer and neural network visualizer modules, it is required to put these modules in the framework to show them on the computer screen. The framework called "SFML" would be very useful for such an application. "SFML" is a graphical multimedia framework designed in C++ programming language to create interactive applications. A class named "Window" class is created to add some sort of customization to the framework Fig XI.

The class will provide a list of adapters which can be used to render any module we have designed so far. However, in order to add the modules to the "Window class", a pattern called "adapters" is designed. Since we have two modules to draw, two adapters are need to be created; one for neural network visualizer and the other one for graph visualizer. When those adapters passed into a "Window" class using "addAdapter()" method and "startWindow()" method, the window appears on the screen showing both of those visualizer modules perfectly.

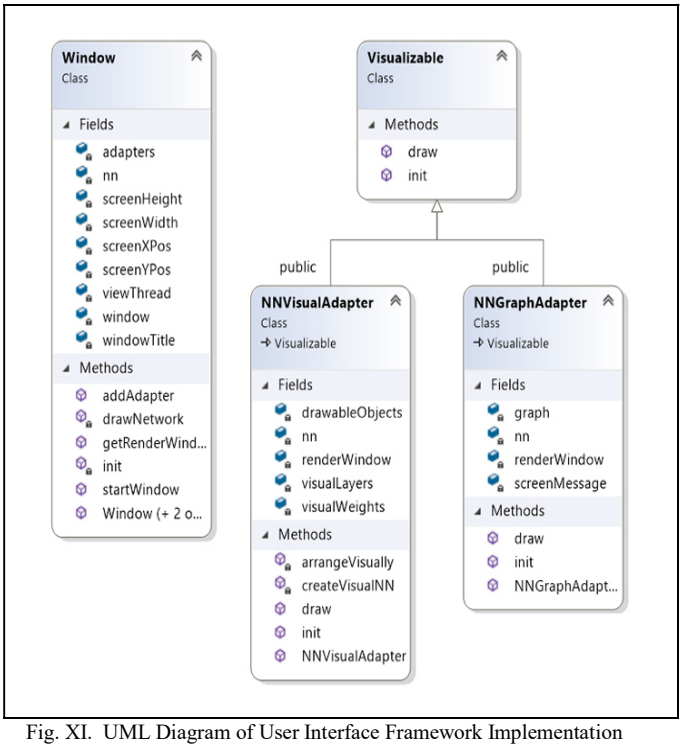

Fig. XI. UML Diagram of User Interface Framework Implementation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

G. *End-User Implementation in Code*

One of the basic objective to this paper is to create a tool to build, calculate, train, test and visualize a neural network. Taking into consideration that there are many means of implementing such a complex architecture to be used in the application of this paper. However, it is important to insure which method is the best for creating this tool, and that is deciding the end-user implementation.

End-user implementation is the code that we decide how the users will call the APIs of our tool. End-user implementation code is given in Fig XII.

First, the user starts with building the neural network. To do that, with previously created builder design pattern, user can call all the necessary addition methods by chaining them in a sequence. In each part of the sequence, they can add layers and each layer can be injected with a dependency of the activation function. Adding activation function to each layer changes the default activation function of the neurons in that same layer. In the end, the last chain learning method can be added with proper cost function instance. In our case, it is Gradient Descent learning with MSE cost function.

After adding the datasets to the code, we can create the two adapters for visualizers. In this specific implementation, two separate windows for demonstrating actions in a neater way are created. Whenever we start the windows, there are three various threads run independently. First one is for background ANN calculations, te second is for ANN visualizer tool, and the the third one is for graph visualizer tool.

```
int numberOfInputs = 4;
int numberOfHiddenLayerNodes = 8;
int numberOfHiddenLayerNodes2 = 8;
int numberOfOutputs = 1;
NeuralNetwork* nn = (new NeuralNetwork::Builder())
        ->addLayer(new Layer(numberOfInputs))
        ->addLayer(new Layer(numberOfHiddenLayerNodes, new ReLU()))
        ->addLayer(new Layer(numberOfHiddenLayerNodes2, new ReLU()))
        ->addLayer(new Layer(numberOfOutputs, new ReLU()))
        ->setLearningMethod(new GradientDescent(new MSE()))
        ->build();
DataSet* trainingDataSet = new DataSet(
                        "data4.data",
                        numberOfInputs,
                        numberOfOutputs
                        );
DataSet* testingDataSet = new DataSet(
                        "data4Test.data",
                        numberOfInputs,
                        numberOfOutputs
                        );
NNVisualAdapter* nnVisualAdapter = new NNVisualAdapter(nn);
NNGraphAdapter* graphAdapter = new NNGraphAdapter(nn, new
FullRange(170));
Window* nnWindow = new Window(
                        sf::Vector2i(700, 600),
                        sf::Vector2i(20, 50),
                        "Neural Network Structure"
                        );
nnWindow->addAdapter(nnVisualAdapter);
Window* graphWindow = new Window(
                        sf::Vector2i(800, 600),
                        sf::Vector2i(720, 50),
                        "Cost Graph"
                        );
graphWindow->addAdapter(graphAdapter);
// Starts new view threads!
nnWindow->startWindow();
graphWindow->startWindow();
double learningRate = 0.03014;
int numberOfIterations = 4500;
nn->train(
        trainingDataSet,
        learningRate,
        numberOfIterations
);
nn->test(testingDataSet);
```

Fig. XII. End-User Implementation Code

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

The moment the training is started, both visualizers start to replicate the state of the original neural network and try to catch up with the speed of background thread without causing any delay or lag. The final step in the implementation was to start testing which can be done easily using the test"datasetInstance" command.

H. Testing

In the end, the framework product is compiled as an executable file with a data file containing training and testing data. When the file is run, it will be able to see two windows in one of which illustrated the animated version of the neural network (Fig. XIII) and the other one contained the graph showing the convergence of the network as the time goes and training finishes.

After training finished, the testing phase completed with 100% correct result showing that have completed the task with great success. As an answer to the initially given problem, we can say 2 hidden layers with 8 neurons in each with ReLU activation function is the best configuration for solving the problem.

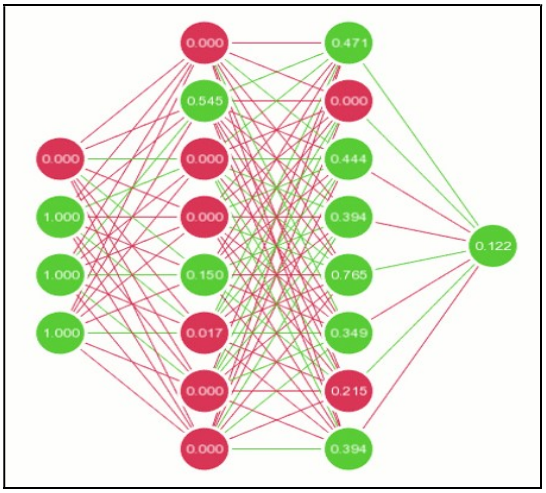

Fig. XIII. XOR Solving ANN Structure Captured from Animation

V. ANN AT ACCIDENT RATES ESTIMATION MODELLING USING CUTOMER MADE FRAMEWORK

Custom neural network framework is already discussed in details in Section IV and it is time to use it. This section concentrates on the properties of the accident rates estimation predicting neural network which is more about the core objective for this paper than how the network itself works.

After many trial and errors, the conclusion was that the best configuration for the neural network in this application of accident rate prediction is to have two hidden layers with sixteen neurons in each.

Since we are trying to find out “Rate of Accident" which is single digit, the output will only consist of single neuron whereas the input layer contains twenty-three neurons – one for each feature. Fig. XIV demonstrates the state of the ANN model during the training session. All those links connecting neurons are the weights, and are shown in different gradient colors between red and green. More red means the value is more negative and greener means the value is more positive.
Training of such a huge neural network took more than 8 hours with the "core i7" hardware. The number of iterations was close to million. Another important parameter was the learning rate. After several trials, it was found that the network performs the best when it is set to 0.03.

Convergence pattern was a little bit different than initial expectation. During the training process, first few thousands iterations did not show any inclination to the convergence, as cost function value should be closer to zero. However, after 250,000 iterations we start to see some progress towards zero. In Fig. VX, it clearly shows that a trend starts to emerge toward down. It is important to highlight about the chart shown in the figure that each value in the x-axis stands for 220 samples in the training process.

The interesting part was what happened after approximately 800,000 cases which are shown from approximately 3600 in the chart. As you can see after almost definite convergence the networks start to deviate and nearly forgot all the training information and act completely randomly for a few thousand samples. But eventually, it rapidly converges and diminishes the cost function result to almost 0.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

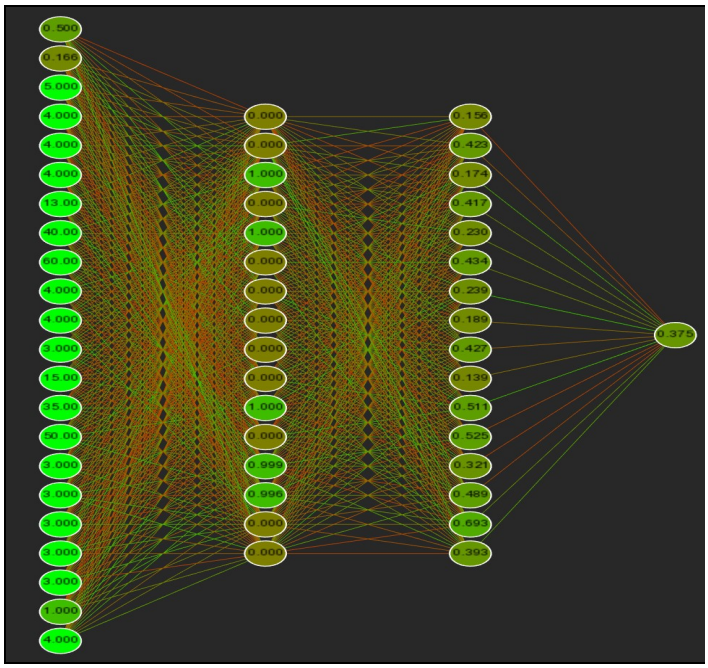

Fig. XIV. ARE Solving ANN Structure Real Time Captured Image

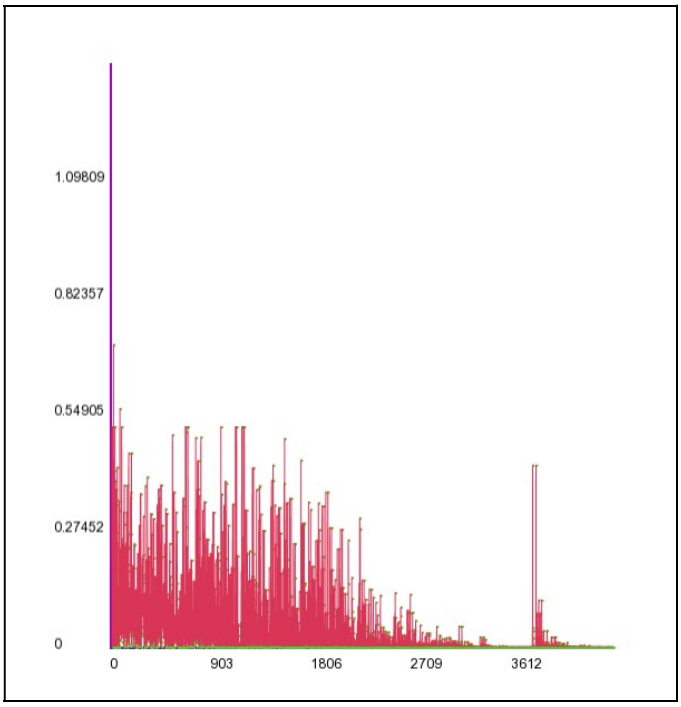

Fig. XV. Full Range Graph

VI. RESULTS AND DISRUPTION

According to the research done on ARE using Fuzzy c-Means Clustering used 8 cases were used to test and validate the model. The same test cases are used in this paper. After all output data is collect for accident rate prediction using ANN, standard deviation was used also to determine the validity of the technique. Using simple comparison, it is easy to recognize which technique is more efficient.

Table III represents the results from final test cases, which is the primary goal of the research. The aim was to predict the accident rates estimation with ANN model as accurately as possible.

From Table III, it can be seen clearly see that the average standard deviation of the error in Fuzzy c-Means Clustering method is %28.68 which gives validity of %71.32 for the given 8 test cases. Whereas artificial neural networks perform with %77.60 validity.

TABLE III.
STANDARD DEVIATION COMPARISON BETWEEN ANN AND FCM CLUSTERING METHODS

Case number	Standard deviation	
	ANN Method	FCM Clustering
1	0.1891	0.3536
2	0.3115	0.2828
3	01609	0.4243
4	0.2530	0.7071
5	0.1228	0.1414
6	01364	0.1414
7	0.3913	0.2121
8	0.2273	0.0318
Average	0.2240	0.2868

From what we have seen during the comparison, we can summarize that Artificial Neural Network method is proved to be performed better than Fuzzy c-Means Clustering method for the given dataset and test cases wish represent very difficult and complicated nonlinear problem. Since the difference between them is not that much high, one can expect that with different dataset, ANN framework will return drastically distinct results. But, if we take into consideration the successful obtained results and the uncertainty of the collected data with the extremely high non-linearity nature of the problem, the results can be improved if the survey questioner is improved and the survey cover higher number of workers with different calibers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 July 2021 doi:10.20944/preprints202107.0016.v1

	FW2	FW3	FW4	FW19
	Weight/Height	Optical status	Hearing Ability	Level of Training
Minimum	0.53617	0.785854	0.756116	0.864666
Maximum	0.72715	0.4851	0.516099	0.437153
Diff	-0.19098	0.300754	0.240017	0.427513

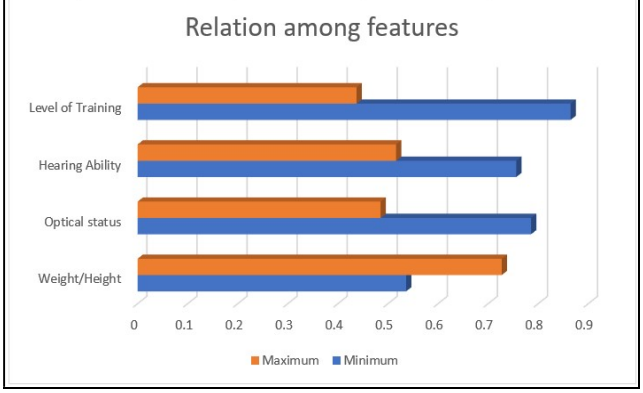

Fig. XVI. Correlation Among Multiple Features in One Case

	FW14	FW15	FW16
	Indoor Work	Office Work	Outdoor Work
Minimum	0.364731	0.708141	0.476297
Maximum	0.800742	0.550991	0.719604
Diff	0.436011	-0.15715	0.243307

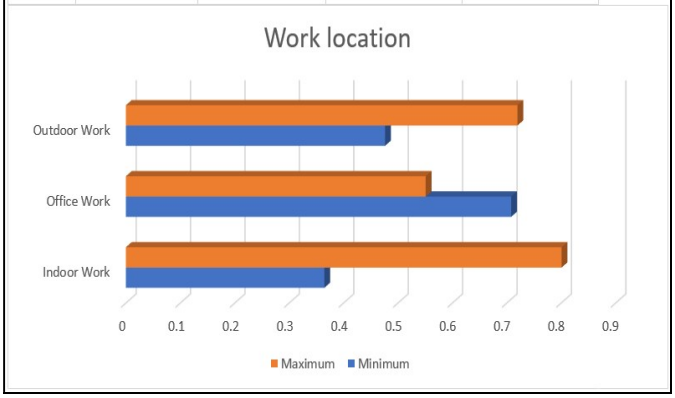

Fig. XVII. Correlation Between Work Location Features

Running the different cases one by one usually does not make a lot of sense to us, however, keeping multiple variable constant and adjusting only one of them gives us very interesting results. It is a great use case for the artificial neural network model. Although its design purpose is to predict what the accident rate will be for an individual, we can virtually create a human model and try to identify which features plays the most important role at accident rates. Fig. XVI below demonstrates one example of such an analysis.

This is a small presentation of what happens in case all the features are kept constant and only one variable is adjusted to its minimum and maximum. I took one case and repeated the same action for all the features and the grouped them according to their category. As the figure demonstrates clearly, for this specific virtual case, the level of training causes an enormous reduction in the rate of accidents from proximally 0.86 to 0.43 to. Features like hearing ability, optical status, weight/height ratio also affect the accident rates quite considerably as shown from the results. Another interesting observation, and it makes sense too, that overweigh increases the rate of accident for this virtual case.

Different chart can be obtained and illustrated in Fig. XVII. This time the study examines the changes in accident rate for the virtual case with respect to three different work locations.

The outcome results show that when indoor work is set to the maximum value the accident rate drops drastically from approximately 0.8 to 0.35, this reduction is more than half. Although outdoor work has a similar effect on accident rate estimate, the ratio of the change is less than the what obtained in indoor work. On the other hand, the results obtained for office work give the reverse effect as minimum accident rate can achieved for this virtual case if this person works only at office, and vice-versa.

Eventually, this is the result achieved from only one case. Further research can improve the results and help to find many realistic values. Having so many correlations for such non-linear model brings lots of surprising results. Thanks to the results of this paper, future researches can obtain much more interesting topic.

VII. CONCLUSION

To summarize, making investigations about the differences between various methods to solve the same problem opens new doors for further research topics. The paper tried to create an artificial neural network model trained with backpropagation gradient descent method using mean squared error cost function with rectified linear unit activation function to solve high nonlinear problem called accident rates estimation. The mentioned problem has already been solved in only one of the previous prior studies using fuzzy c-means clustering method to some extent. The question asked in this paper was which one of these technique would perform better if we gave them the same initial conditions which are, in this case, the dataset for training and testing.

To achieve the goal, fully functioning neural network framework was created with advanced visualizing tools to train the model. Then to ensure that the network adapts to the model and predicts reasonable results, the framework was tested with simple task. After solving the task, the ANN tool was ready for solving the accident rates estimation problem.

Creation of the neural network completed after trial and errors process to find the optimum configuration for it. Resulted network passed through nearly a million iterations of training sample cases to reach convergence. When it was ready, given test cases were introduced to the trained neural network for prediction. Results were impressive, taking into consideration that validity obtained by ANN was equal to %77.6 whereas the validity using fuzzy c-means clustering method was equal to %71.32. That is mean %6.28 percent more accurate result achieved by the artificial neural network over fuzzy c-means clustering.

The successful approach discussed in this paper can be used widely in insurance companies, and their clients too, as a good indicator for better estimate the insurance fees depending on actual prediction of accidents for laborers based on direct factors related to the laborers themselves instead of using historical data that, i believe, would not be valid for the future accident estimations as the laborers may not be working in the same company at the time of accident when happen. Another practical application for this paper can be in industrial companies and construction contractor companies to provide safer working conditions for laborers by focusing on the human factors and the effective features causing accidents to improve it in order to insure maximum decreasing in the accident rate of labors without wasting efforts and with minimum possible cost. Hopefully, further research will reveal much more efficient methods to solve the to reduce the accident rate for better safety for workers.

REFERENCES

A. Lukacova, F. Babic and J. Paralic, "Building the prediction model from the aviation incident data.in applied machine intelligence and informatics", (SAMI), IEEE 12th International Symposium on, pp. 365-369, 2014.

A. J.-P. Tixier, M. R. Hallowell, B. Rajagopalan and D. Bowman, "Application of machine learning to construction injury Prediction", Automation in Construction, vol. 69, pp. 102-114,2016. http://dx.doi.org/10.1016/j.autcon.2016.05.016.

A. P. Akgüngör and E. Doğan, “An artificial intelligent approach to traffic accident estimation: Model development and application,” Transport, vol. 24, no. 2, pp. 135–142, 2009.

Abbas, M. “Comparison of weighted and simple linear regression and artificial neural network models in freeway accidents prediction (Case study): Qom & Qazvin Freeways in Iran”. Proc. of Second International Conference on Computer and Network Technology. Tehran, Iran: IEEE, 2010.

Amirasln Bakhshil Muhammad M.A.S. Mahmoud, Leyla Muradkhanli, “Accident rate estimate modeling based on human factors using artificial intelligent technique”, 1st National Students Scientific Conference, Process Automation and information, April, 2019.

Ashok Kumar Bansal, “Industrial accidents and their prevention: A case of Satluj Jal Viduat Nigam Limited, Shimla, Himachal Pradesh, intellectual property rights”, vol. 4 Issue 3, 2016. ISSN: 2375-4516, DOI: 10.4172/2375-4516.1000171.

Borja García de SotoBorja García de SotoAndreas Bumbacher Markus DeubleinMarkus DeubleinBryan AdeyBryan Adey, “Predicting road traffic accidents using artificial neural network Models”, Infrastructure Asset Management, April 2018. DOI: 10.1680/jinam.17.00028.

Choi, S.D. (2006). “A survey of safety roles and cost of injuries in the roofing contracting injury”, Journal of Safety, Health and Environmental Research, Vol. 3, pp. 1-20.

Costarelli D., Croitoru A., Gavrilut A., Iosif A., Sambucini A.R., ”The Riemann-Lebesgue Integral of Interval-Valued Multifunctions, Mathematics 2020, 8, 2250; doi:10.3390/math8122250.

Douglas A. Wiegmann “A human error analysis of commercial aviation accidents using the human factors analysis and classification system report”, National Technical Information Service, 2001.

Douglas A. Wiegmann, Scott A. Shappell, “Human factors analysis of post-accident data: Applying theoretical taxonomies of human error”. The International Journal of Aviation Psychology, 7(1), 67-81, 2005.

Eunsuk ChoiEunsuk ChoiGyeong-Suk JeonGyeong-Suk JeonWon Kee LeeYoung Sun Kim, “The prediction of industrial accident rate in korea: A Time Series Analysis”, Korean Journal of Occupational Health Nursing, vol. 25(1), pp 65-74, February 2016 DOI:10.5807/kjohn.2016.25.1.65.

Evanoff B, Abedin S, Grayson D, Dale A, Wolf L, et al. “Is disability underreported following work injury?” Journal of Occupational Rehabilitation vol. 12: pp139-150, 2002.

F. Pereira, F. Rodrigues and M. Ben-Akiva, "Text analysis in incident duration prediction", Transportation Research Part C: Emerging Technologies, vol. 37, pp. 177-192, 2013.

F. Rezaie MoghaddamSh. AfandizadehMojtaba ZiyadiMojtaba Ziyadi, “Prediction of accident severity using artificial neural networks”, International Journal of Civil Engineering, vol. 9(1), pp 41-49, March 2011.

Grabish, M. – Fuzzy integral in multicriteria decision making, Fuzzy Sets and Systems, 69 (1995), 279-298.

Handley W “Industrial safety handbook”. London, McGraw-Hill,. 1977.

Hidetake Sakuma, “Strategic accident prevention with applied human factors theories”, International Railway Safety Conference Tokyo 2002.

Jesús Domech Moré, Ari Sauer Guimarães, Geraldo Bonorino Xexéo, Ricardo Tanscheit. “A fuzzy approach to the evaluation of human factor in ultrasonic nondestructive examinations”. Journal of Industrial Engineering International, Vol. 3, No. 5, 41-52, July 2007.

Kay Yong, Thomas Wang, “Human Factors analysis and methodology of the SQ006 accident”, Aviation Safety Council of Taiwan, 2007.

Khair S. Jadaan, Muaath Al-Fayyad, and Hala F. Gammoh, “Prediction of road traffic accidents in Jordan using artificial neural network (ANN), Journal of Traffic and Logistics Engineering, vol. 2, No. 2, June 2014.

Manar M. Sabry & Magdi S. Mahmoud “Modeling of human accident intervention and accident rate prediction using fuzzy techniques”, Systems Analysis Modelling Simulation, viol 42(12), pp 1783-1805, 2002, DOI: 10.1080/911178269, http://dx.doi.org/10.1080/911178269.

Mohamed MAS Mahmoud, Sorin Monuu “Accident rates estimation modelling based on human factors using fuzzy c-means clustering algorithm”, Journal of Communication and Computer - World Academy of Science, vol.9, pp 1298-1309 2012.

Mohd Zakwan Bin Ramli, “Development of accident prediction model by using artificial neural network (ANN)”, A project report submitted in partial fulfillment of the requirement for the award of the Degree of Master of Civil Engineering Faculty of Civil and Environmental Engineering University Tun Hussien Onn, Malaysia May 2011.

P. Hämäläinen, J. Takala and K. Saarela, "Global estimates of occupational accidents", Safety Science, vol. 44, no. 2, pp. 137-156, 2006.

PEYTON, R. X. “Construction safety practices and principles”. Van Nostrand Reinhold, New York, 1991.

Pham, T.D., Brandl, M., Nguyen, N.D., Nguyen, T.V. – Fuzzy measure of multiple risk factors in the predictions of osteoporotic fractures, Proceedings of the 9-th WSEAS International Conferences on Fuzzy Systems (FS’08), 2008, 171-177.

Sobhan SarkarSobhan SarkarAtul PatelSarthak MadaanSarthak MadaanJhareswar MaitiJhareswar Maiti, “Prediction of occupational accidents using decision tree approach”, Project: UAY: An MHRD Sponsored Project - Data Analytics in Industrial Safety, Conference: INDICON-2016 (IEEE)At: IISC Bangalore, December 2016, DOI: 10.1109/INDICON.2016.7838969.

Strong, M.J “Human Factors in Occupational Injury evaluation and control”. Handbook of Human factors. John Wiley and Sons Inc, N.Y. (1987).

T.A Yusuf, S.O Ismaila, S.I Kuye, O.D Samuel, “Evaluation of the cost and effect of Industrial Accidents”, European Journal of Scientific Research, Vol. 132 No 2, pp.184-190, May, 2015, ISSN 1450-216X / 1450-202X, http://www.europeanjournalofscientificresearch.com.

Wilson, H.A “Organisational behaviour and safety management in construction industry”, Construction Management and Economics, Vol. 7, pp. 303-319, 1989.

Y.-C. Chiou, “An artificial neural network-based expert system for the appraisal of two-car crash accidents,” Accident Analysis and Prevention, vol. 38, no. 4, pp. 777–785, 2006.

Youhee Choi, Jeong-Ho Park. Byungtae Jang , “Developing safety checklists for predicting accidents”, 2018 International Conference on Information and Communication Technology Convergence (ICTC), DOI: 10.1109/ICTC.2018.8539652, Oct. 2018.

Professor Dr. Muhammad M.A.S. Mahmoud, Egyptian, Received the B.S. degree in Electrical Engineering from Cairo University and the M.Sc. degree from Kuwait University. First Ph.D. degree from Transilvania University of Brasov, Romania in IT and Computer. Second PhD Degree in Electrical Power system and Machine, Cairo Univ. Egypt. He occupies a position of Professor in Process Automation Engineering Department, Baku Higher Oi School, Azerbaijan. His current research interests in Fuzzy and Artificial Neural Network Techniques application include power delivery, protection reliability, control, safety, building automation, smart city, and energy management. Prof. Dr. Muhammad is IEEE Senior Member (SM) since 2001 and TFS –IEEE Reviewer 2016.