# Survival Probabilities of a Markov Dependent Insurance risk model and its Stochastic Analysis using Machine Learning Algorithm

**G Jagath Prasad Sreedhar[1]* and Dr. K K Suresh[2]**

*Department of Statistics, Bharathiar University, Coimbatore, Tamilnadu, India.*

*Abstract* - A numerical solution of the survival probability of a renewal time-dependent insurance risk model with Erlang(2) claim sizes is calculated using Legendre orthogonal polynomials using a machine-learning algorithm called Extreme Learning Machine(ELM). The hidden layer in the ELM algorithm is replaced with Legendre orthogonal polynomials for solving a system of linear equations and the parameters of the artificial neural network are calculated using the ELM algorithm. We compared the superiority of a dependent Erlang(2)insurance risk model against a dependent exponential insurance risk model and also tested the validity and reliability of our suggested Legendre polynomial neural network extreme learning machine (Lnn-ELM) technique by finding a numerical solution to the Erlang(2) model and compared it with its exact numerical solution using a multi-layer perceptron artificial neural network(MLPANN). The findings generated by our suggested Lnn-ELM model reveal very high accuracy when compared to the existing method.

***Keywords:*** Survival probability, Dependent Erlang (2) risk model, Artificial Neural Networks in insurance, Legendre Neural Network

## 1. Introduction

Ruin (risk) theory is a popular topic in finance and mathematics since it is the most significant element of actuarial mathematics [4] [12] [16]. For constructing mathematical models in describing risks involved in business, ruin theory uses probability theory along with the stochastic process. Estimating risk by finding ruin or survival probabilities plays a significant role in formulating strategies and avoiding bankruptcy for the insurance business. When the probability of survival is less or ruin probability is more, that is an indicator that the insurance firm is on the edge of bankruptcy and the company is forced to take steps such as reinsurance mechanism or increasing premiums, or raising extra capital from other ways. Although the probability of survival does not alone precisely reflect the probability that an insurance firm would go out of business in the coming future, the insurance company can formulate different policy combinations by looking at the survival probability.

A great extent of research has been going on in this field to find the ruin probabilities by developing different insurance risk models [1]- [3], [5]- [8]. All the models developed by researchers all over the world face the question" How much insurance firms may suffer insolvency". By calculating ruin probability, this question can be answered by making it not only an important indicator for ensuring the insurance firm's regular functioning but also a quantitative criterion for the insurance company's survival by controlling the risks associated with it [4]. However, finding the exact value of ruin probability is difficult due to the randomness in actuarial calculations.

So many insurance risk models are developed on numerous assumptions, such as the standard Poisson Risk models (Classical Cramer-Lundberg risk model) [18], Erlangian dependent and independent risk models [6]-[8], Sparre-Andersen models [1] [3], etc... Usually, in all these models, the ruin probability satisfies an integro-differential equation obtained and in the majority cases, the absence of an exact solution of these integro-differential can be identified. From the time beginning itself, the Monte-Carlo algorithm [26], was used to simulate the numerical-value of survival probability or its upper bound. In our paper, we tried to develop an accurate numerical solution to survival probability using the machine learning method called Legendre neural network extreme learning machine (Lnn-ELM) technique.

The Lundberg-Cramer inequality (Classical Risk Model) and the integro-differential equation are satisfied by the survival probability. In general, these models only estimate an upper bound or lower bound of probability of ruin or probability of survival for different distributions. However, a numerical solution of the probability of survival is frequently required in actuarial research and the designing of insurance portfolios. While considering different integro-differential equations involved in these models, their numerical solutions can be solved in a variety of ways. Here we tried to get the solution numerically for the survival probability for the dependent renewal risk model **[15]** with Erlang(2) claim distribution and exponential inter-arrival times using a machine learning tool from artificial intelligence called Lnn-ELM [13].

Zhou et al[30] proposed an enhanced artificial neural network (ANN) method by using a trigonometric function as an activation function or basis function in the ELM architecture to find the numerical value of the ruin probability in the classical insurance model with an exponential claim size distribution. The aim of this research paper is to use an artificial neural network with Legendre orthogonal polynomials to provide an accurate numerical solution for the survival probability for renewal risk models with dependence when claim size distribution is Erlang(2). A comparison of the exact solution is done with a popular dependent renewal risk model with exponential claim size distribution.

Also, a multilayer perceptron artificial neural network (MLPANN) [24] is developed using the MATLAB platform with machine learning concepts for comparison with the Lnn-ELM model. MLP is a part of a feedforward AI network, working with a backpropagation algorithm. MLP has one input layer, one output layer, and one or more hidden layers, with many neurons stacked on top of each other. In order to minimize the cost functions, backpropagation is a learning mechanism that allows a multi-layer perceptron to iteratively adjust weights in a network. [21] - [25]

The following sections of the paper, Section 2 methodology used discusses the method used for model development, Section 3 illustrates Results and Discussions that describes the validation of numerical examples along with a comparison of the Erlang(2) model with Exponential model and ANN model, with conclusion and future work in section 4.

## 2. Methodology

The workflow chart relating to model development and validation is given below in Fig. 1.

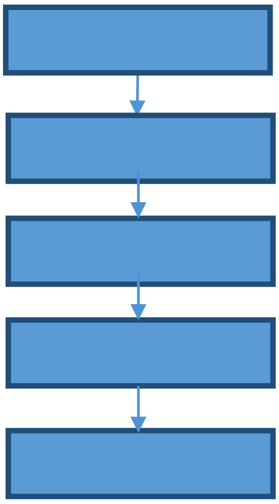

**Fig. 1**. Work flow chart

## 2.1 *Motivation*

Yang and Tseng [28] proposed the method of approximating functions using Legendre polynomial Neural Network (Lnn). The Legendre orthogonal polynomial with a single layer neural network method for calculating the survival probabilities has the following advantages:

a) Weights for the output layer only assigned in Lnn.

b) Simple computation and implementation.

c) A system of linear equations, the output-weights of the mathematical problems can be calculated using inverse matrix in the generalized form, resulting in the speeding up of iteration.

d) The Legendre neural network approach has a higher calculation accuracy than Multi-Layer Perception.

e) Initializing a Legendre Expansion in the input pattern eliminates the hidden layer.

f) No optimization procedure is needed.

## *2.2 Extreme Learning Machine Algorithm*

For $K$ different samples $(x_j, p_j) \in \Re^2 \ \forall \ j. = 1,2,3....,K$, the artificial neural network with $N$ hidden layer neurons can be represented in the mathematical form[13],

$$\sum_{i=1}^{N} B_i . f(w_i x_j + b_i) \ , \ j. = 1,2,3,4,...,K \quad (1)$$

where $B_i$, the weight matrix in the output which is connected the $i^{th}$ hidden-layer node and the output node, $w_i$ be the weight in the input , which bonds the $i^{th}$ hidden layer node with an input node, and $b_j$ represents the bias. The error is zero when we use $N$ hidden neurons. Hence, we have

$$\sum_{i=1}^{N} B_i . f(w_i . x_j + b_i) = q_j \quad forevery \ \ j = 1,2,3,4,...,K \quad (2)$$

Rewriting (2) as a matrix form; $H\,B = Q$ (3)

Here the output matrix $H$ of the hidden-layer , given by

$$H = \begin{bmatrix} f(w_1 x_1 + b_1) & f(w_2 x_1 + b_2) & f(w_3 x_1 + b_3) & ... & f(w_N x_1 + b_N) \\ f(w_1 x_2 + b_1) & f(w_2 x_2 + b_2) & f(w_3 x_2 + b_3) & ... & f(w_N x_2 + b_N) \\ f(w_1 x_3 + b_1) & f(w_2 x_3 + b_2) & f(w_3 x_3 + b_3) & ... & f(w_N x_3 + b_N) \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ f(w_1 x_M + b_1) & f(w_2 x_M + b_2) & f(w_3 x_M + b_3) & ... & f(w_N x_N + b_N) \end{bmatrix} \text{of order } M \ XN,$$

$$B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ . \\ . \\ . \\ B_N \end{bmatrix}, \text{ and } \ Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ . \\ . \\ . \\ q_M \end{bmatrix}$$

As previously stated, the input-weights $w_i$ with hidden- biases $b_i$ can be generated randomly. .So the output weight is same as the solution of a system of linear equations (3)

The minimum least square solution of the linear system of (3) given by $B = H^{\dagger}Q$ , (4)

In (4), $H^{\dagger}$ is the inverse matrix in the generalized form called Moore-Penrose inverse matrix of $H$

### *2.3 **Legendre polynomial neural network extreme learning machine (Lnn-ELM) algorithm for the survival probability.***

Here, we describe a single-layer Lnn-ELM algorithm along with its architecture, and a numerical method is proposed by using Legendre orthogonal polynomials in solving the survival probabilities of the dependent risk model with Erlang (2) claim sizes.

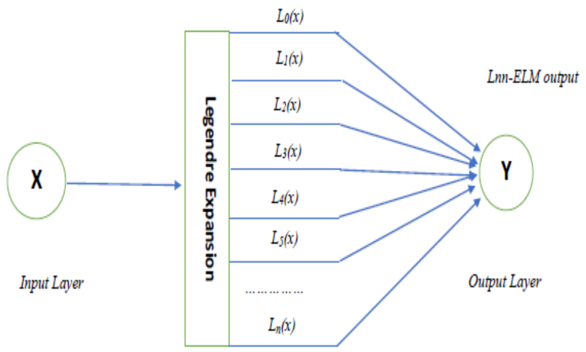


**Fig. 2** Network Architecture

The Legendre neural network algorithm [11][17] is comprised of an input node, a functional expansion block based on Legendre polynomials, and an output node, as shown in Figure 2. Here the hidden layer is removed by adding Legendre polynomials to the input pattern. Legendre polynomials, [22] consist of a collection of orthogonal polynomials produced by solving the Legendre Differential Equation.

The first and second Legendre polynomials are,

$$L_0(u) = 1, \quad L_1(u) = 0$$

The following well-known recursive formula could be used to construct higher-order Legendre polynomials.

If $L_n(u)$ be the nth order Legendre polynomial, then

$$L_{n+1}(u) = \frac{(2n+1)uL_n(u) - nL_{n-1}(u)}{n+1}, \ \forall n \geq 1 \ \ and \ -1 < u < 1 \quad \text{is the argument} \tag{5}$$

**Remark 1**. The following are some of the benefits of using Legendre polynomials as activation functions in our proposed algorithm:

a) Orthogonal property of Legendre polynomials

$\int_{-1}^{1} L_i(u)L_j(u)du = \frac{2}{2i+1}\delta_{ij}$, where $\delta_{ij}$ is the Kronecker delta. By considering a interval [a, b], the function as such is represented as Legendre linear polynomials.

b) Bias functions in Legendre can differentiate and it satisfies $L'(u) = L(u)P$, where $P$ matrix is given by

$$P = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots & 0 & 1 \\ 0 & 0 & 3 & 0 & 3 & 0 & 3 & \dots & 3 & 0 \\ 0 & 0 & 0 & 5 & 0 & 5 & 0 & \dots & 0 & 5 \\ 0 & 0 & 0 & 0 & 7 & 0 & 7 & \dots & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 0 & \dots & 0 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 11 & \dots & 11 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 2M-1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}_{(M+1)(M+1)} \quad (6)$$

$L(u) = \left\{L_0(u), L_1(u), L_2(u), \ldots, L_M(u)\right\}$ and

$L'(u) = \left\{L_0'(u), L_1'(u), L_2'(u), \ldots, L_M'(u)\right\}$

### *2.4 Model 1 - Survival probability in the dependent renewal risk model with Exponential claim size distribution*

This is the popular dependent insurance risk model **[20]** in which the duration between two claim occurrences influences the next claim size distribution, and claim sizes follow exponential distribution. In a dependent exponential insurance risk model,

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i, t \geq 0$$

where $\{U(t)\}_{t\geq 0}$ is the surplus, $X_i$ is the claim sizes , $u$ is positive initial-capital and $c$ is the premium income. $N(t)$ is the claim number which is a Poisson process with parameter $\lambda$. $\{X_1, X_2, X_3, \ldots, X_{N(t)}\}$ is a sequence of non-negative, independent i.i.d random variables with $X_i$ representing the $ith$ claim. $\{T_1, T_2, T_3, \ldots, T_n\}$ is a sequence of nonnegative i.i.d random variables, where $T_i$ denotes the time between $(i-1)th$ and $i\ th$ claims and $\{M_i, i = 1,2,3,4,\ldots\}$ is a sequence of iid exponential random variables. If $T_i < M_i$, then $X_i \sim F_1(x)$ and if $T_i > M_i$, then $X_i \sim F_2(x)$, where $M_i$, $F_1$ and $F_2$ are exponential distributions with parameters $\delta$, $\alpha$ and $\beta$ respectively. The pdf of an exponential distribution is $\beta\, e^{-\beta x}$ for $x \geq 0$. Also $M_i's$ are independent of $T_i's$ and $F_i's$, $i = 1,2,3\ldots$

The expected positive net-profit condition is, $\frac{c}{\lambda} > P(M > T).E(F_1 \setminus M > T) + P(M \leq T).E(F_2 \setminus M \leq T)$

The probability of ruin is given by $\Psi(u) = P\{Ruin \setminus U(0) = u;\} = P\{T < \infty\}$.

The survival probability is $\Phi(u) = P\{U(t) \geq 0,\ forevery\ t \geq 0\} = P(T = \infty)$.

The probability of survival for the risk model with Exponential claim size distribution [2] satisfies the following integro-differential equation,

$$\frac{c}{\lambda}\Phi''(u) = \left(2 + \frac{\delta}{\lambda}\right)\Phi'(u) - \left(\alpha + \frac{\delta}{c} + \frac{\lambda}{c}\right)\Phi(u) + \left(\alpha^2 + \frac{\lambda\alpha}{c}\right)\int_0^u \Phi(u-x)\, e^{-\alpha x}\, dx$$

$$+ \frac{\delta\beta}{c}\int_0^u \Phi(u-x)\, e^{-\beta_2 x}\, dx$$

$$\Phi(0) = \frac{\lambda}{cs_0}\left\{\left(2 + \frac{\delta}{\lambda}\right) - \left(1 + \frac{\lambda}{c\alpha}\right) - \frac{\lambda}{c\beta}\right\}$$

The survival probability is given by $\Phi(u) = 1 + Pa_2 e^{R_1 u} + Pa_3 e^{R_2 u}$ where

$$P = \frac{\lambda}{cs_0}\left\{1+\frac{\delta}{\lambda}-\frac{\lambda}{c\alpha}-\frac{\lambda}{c\beta}\right\}, \quad a_1 = \frac{\alpha.\beta}{R_2 R_1}, \quad a_2 = \frac{(\alpha+R_1)(\beta+R_1)}{R_1(R_1-R_2)}, \quad a_3 = \frac{(\alpha+R_2)(\beta+R_2)}{R_2(R_2-R_1)}$$

$s_0$ is the positive root and $R_1$, $R_2$ the two negative roots [2]

### 2.5 *Model 2 - Survival probability of the dependent renewal risk model with Erlang (2) claim size distribution*

For a dependent Erlang(2) risk model **[15]**, the surplus of an insurance company $\{U(t)\}_{t\geq 0}$ with premium income $c$ per unit time and claim sizes $X_i$ is

$$U(t) = u + ct - \sum_{i=1}^{N(t)} X_i \, , t \geq 0 \tag{7}$$

Where, $u$ is the initial capital, $N(t)$ is a Poisson process with parameter $\lambda > 0$, and $\{X_1, X_2, X_3, ....., X_{N(t)}\}$ is a sequence of non-negative, independent i.i.d random variables with $X_i$ representing the $ith$ claim. $\{T_1, T_2, T_3, ...., T_n\}$ is a sequence of non-negative i.i.d random variables, where $T_i$ denotes the time between $(i-1)th$ and $i\ th$ claims. If $T_i < M_i$, then $X_i \sim F_1(x)$ and if $T_i > M_i$, then $X_i \sim F_2(x)$, where respectively $F_1$ and $F_2$ are two Erlang distributions with parameters $\beta_1$ and $\beta_2$ and the pdf is given as $\beta^2 x e^{-\beta x}$ for $x \geq 0$, $\beta$ being integer. Also $M_i's$ are independent of $T_i's$ and $F_i's$, where $i = 1,2,3,4....$

Here $N(t)$ follows a Poisson process with parameter $\lambda$, it is understood that for $i = 1,2,3,4...$, $T_i's$ are distributed as exponential with parameter $\lambda$, a renewal process with exponentially-distributed inter-arrival times can be considered as a Poisson process. We assume that $A_i's$ also distributed exponential distribution with parameter $\gamma$, as $T_i's$ having a threshold at value $A_i$. The expected positive net-profit condition is,

$$\frac{c}{\lambda} > 2\left[\frac{P(A \leq T)}{\beta_2} + \frac{P(A > T)}{\beta_1}\right] \tag{8}$$

By considering the condition that, the first claim inter-occurrence time distributed as exponentially with rate $\lambda$, the probability of survival is denoted by $\Phi(u)$ with an initial capital $u$. Then ultimate ruin probability is denoted by $\Psi(u)$, where $\Psi(u)$ is given as

$$\Psi(u) = P[\,ruin \setminus U(0) = u;] = P[T < \infty] \tag{9}$$

The explicit value of probability of ruin has the form $\psi(u) = 1 - \Phi(u)$, and probability of survival is

$$\Phi(u) = P[U(t) \geq 0, \ forevery \ t \geq 0] = P(T = \infty) \tag{10}$$

The ultimate survival probability $\Phi(u)$ is given by

$$\Phi(u) = P[u + \sum_{i=0}^{n}(cT_i - X_i) > 0 \ \ for\ some\ n \in N]\,,\ u \geq 0 \tag{11}$$

The survival probability for Erlang(2) claim size distribution model [15] satisfies the following integro-differential equation,

$$\frac{c}{\lambda}\Phi''(u) = \left(2+\frac{\gamma}{\lambda}\right)\Phi'(u) - \frac{(\lambda+\gamma)}{c}\Phi(u) + \left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\int_0^u \Phi(u-x)\, x\, e^{-\beta_1 x}\, dx$$

$$-\ \beta_1^2\int_0^u \Phi(u-x)\ e^{-\beta_1 x}\, dx + \frac{\gamma\beta_2^2}{c}\int_0^u \Phi(u-x)\, x e^{-\beta_2 x}\, dx$$

$$\frac{c}{\lambda}\Phi''(u)=\left(2+\frac{\gamma}{\lambda}\right)\Phi'(u)-\frac{(\lambda+\gamma)}{c}\Phi(u)+\left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\int_0^u\Phi(u-x)\,x\,e^{-\beta_1 x}\,dx$$

$$-\beta_1^2[1-F_1(u)]+\frac{\gamma\beta_2^2}{c}\int_0^u\Phi(u-x)\,xe^{-\beta_2 x}\,dx \tag{12}$$

Let $\widetilde{\Phi}(u)$ denotes the Laplace Transform of $\Phi(u)$ and is defined as $\widetilde{\Phi}(u)=\int_0^\infty e^{-su}\,\Phi(u)\,du$. By applying Laplace transform properties, we get,

$$\widetilde{\Phi}(s)=\frac{\frac{c}{\lambda}\Phi'(0)+\frac{c}{\lambda}s\Phi(0)-\left(2+\frac{\gamma}{\lambda}\right)\Phi(0)}{\frac{c}{\lambda}s^2-\left(2+\frac{\gamma}{\lambda}\right)s+\frac{(\lambda+\gamma)}{c}-\frac{\left(\frac{\lambda}{c}+\beta_1\right)\beta_1^2}{(s+\beta_1)^2}-\frac{\gamma\beta_2^2}{c(s+\beta_2)^2}+\frac{\beta_1^2}{(s+\beta_1)}}. \tag{13}$$

The boundary condition of $\Phi(\infty)=1$ and the final-value theorem for Laplace transform yield that,

$$\lim_{u\to 0} s\widetilde{\Phi}(s)=\lim_{s\to\infty}\Phi(u)=\Phi(\infty)=1$$

and hence the boundary conditions at $0$ and $\infty$ represent the limiting case of the Laplace transform as such,

$$\lim_{s\to 0}\frac{\frac{c}{\lambda}\Phi'(0)\,s+\frac{cs^2}{\lambda}\Phi(0)-\left(2+\frac{\gamma}{\lambda}\right)\Phi(0)s}{\frac{c}{\lambda}s^2-\left(2+\frac{\gamma}{\lambda}\right)s+\frac{(\lambda+\gamma)}{c}-\left(\frac{\lambda}{c}+\beta_1\right)\frac{\beta_1^2}{(s+\beta_1)^2}-\frac{\gamma\beta_2^2}{c(s+\beta_2)^2}+\frac{\beta_1^2}{(s+\beta_1)}}=1 \tag{14}$$

By applying the L'Hospitals' rule, we can individually differentiate the numerator and denominator to apply limiting the function, indeed we obtain,

$$\lim_{s\to 0}\frac{\frac{c}{\lambda}\Phi'(0)+\frac{2cs}{\lambda}\Phi(0)-\left(2+\frac{\gamma}{\lambda}\right)\Phi(0)}{\frac{2cs}{\lambda}-\left(2+\frac{\gamma}{\lambda}\right)+2\left(\frac{\lambda}{c}+\beta_1\right)\frac{\beta_1^2}{(s+\beta_1)^3}+\frac{2\gamma\beta_2^2}{c(s+\beta_2)^3}-\frac{\beta_1^2}{(s+\beta_1)^2}}=1 \tag{15}$$

$$\Phi(0)=-\frac{\lambda}{cs_0}\left\{1-\left(2+\frac{\gamma}{\lambda}\right)+\frac{2\lambda}{c\beta_1}+\frac{2\gamma}{c\beta_2}\right\} \tag{16}$$

Where, $s_0$ being the positive root of

$$\frac{c}{\lambda}\Phi'(0)+\frac{c}{\lambda}\Phi(0)\,s_0-\left(2+\frac{\gamma}{\lambda}\right)\Phi(0)=0 \tag{17}$$

Using $(13)$ and $(17)$ and by using limiting conditions of survival probability we have the Laplace transform as,

$$\widetilde{\Phi}(s)=\frac{\left\{1+\frac{\gamma}{\lambda}-\frac{2\lambda}{c\beta_1}-\frac{2\gamma}{c\beta_2}\right\}\left\{\frac{s-s_0}{s_0}\right\}}{\frac{c}{\lambda}s^2-\left(2+\frac{\gamma}{\lambda}\right)s+\frac{(\lambda+\gamma)}{c}-\frac{\left(\frac{\lambda}{c}+\beta_1\right)\beta_1^2}{(s+\beta_1)^2}-\frac{\gamma\beta_2^2}{c(s+\beta_2)^2}+\frac{\beta_1^2}{(s+\beta_1)}} \tag{18}$$

where $s_0$ is the positive root. Let the denominator of $(18)$ be represented by $Dr(s)$ where,

$$Dr(s) = \frac{c}{\lambda}s^2 - \left(2+\frac{\gamma}{\lambda}\right)s + \frac{(\lambda+\gamma)}{c} - \left\{\frac{\left(\frac{\lambda}{c}+\beta_1\right)\beta_1^2}{(s+\beta_1)^2} + \frac{\gamma\beta_2^2}{c(s+\beta_2)^2} - \frac{\beta_1^2}{(s+\beta_1)}\right\}$$

Now let $Dr(s) = \dfrac{\frac{c}{\lambda}N(s)}{(s+\beta_1)^2(s+\beta_2)^2}$ .Here $N(s)$ can be expressed as a polynomial of degree $6$ with leading coefficient $1$ . Let the roots of $N(s)$ are $0,\ s_0,\ R_1,\ R_2,\ R_3$ and $R_4$ which is same as the roots of the denominator of $(11)$ . Therefore, we can express $Dr(s)$ in theform, $Dr(s) = \dfrac{\frac{c}{\lambda}s(s-s_0)(s-R_1)(s-R_2)(s-R_3)(s-R_4)}{(s+\beta_1)^2(s+\beta_2)^2}$
(19)

and from equation $(18)$, the Laplace transform can be expressed in the form,

$$\tilde{\Phi}(s) = B_{s_0}\frac{(s+\beta_1)^2(s+\beta_2)^2}{s(s-R_1)(s-R_2)(s-R_3)(s-R_4)}$$

where $B_{s_0} = \dfrac{\lambda}{cs_0}\left\{1+\dfrac{\gamma}{\lambda}-\dfrac{2\lambda}{c\beta_1}-\dfrac{2\gamma}{c\beta_2}\right\}$ which is a constant depends on the positive root of the $\tilde{\Phi}(s)$ and parameters of the model. Specifically $\tilde{\Phi}(s) = B_{s_0}\chi(s)$ where $\chi(s) = \dfrac{(s+\beta_1)^2(s+\beta_2)^2}{s(s-R_1)(s-R_2)(s-R_3)(s-R_4)}$ which can be obtained by factoring $\chi(s)$ in such a way that $\chi(s) = \dfrac{A_1}{s}+\dfrac{A_2}{s-R_1}+\dfrac{A_3}{s-R_2}+\dfrac{A_4}{s-R_3}+\dfrac{A_5}{s-R_4}$ .

Obviously $A_1,\ A_2,\ A_3,\ A_4,\ A_5$ are constants obtained by limiting the values of $s$ to $0,\ R_1,\ R_2,\ R_3$ and $R_4$ in $\chi(s)$ . That implies,

$$A_1 = \frac{(\beta_1\beta_2)^2}{R_1R_2R_3R_4} \qquad A_2 = \frac{(R_1+\beta_1)^2(R_1+\beta_2)^2}{R_1(R_1-R_2)(R_1-R_3)(R_1-R_4)}, \qquad A_3 = \frac{(R_2+\beta_1)^2(R_2+\beta_2)^2}{R_2(R_2-R_1)(R_2-R_3)(R_2-R_4)}$$

$$A_4 = \frac{(R_3+\beta_1)^2(R_3+\beta_2)^2}{R_3(R_3-R_1)(R_3-R_2)(R_3-R_4)}, \quad A_5 = \frac{(R_4+\beta_1)^2(R_4+\beta_2)^2}{R_4(R_4-R_1)(R_4-R_2)(R_4-R_3)} \qquad (20)$$

The exact expression of the survival probability for Erlang distributed claim sizes is given by

$$\Phi(u) = 1 + B_{s_0}A_2e^{R_1u} + B_{s_0}A_3e^{R_2u} + B_{s_0}A_4e^{R_3u} + B_{s_0}A_5e^{R_4u} \qquad (21)$$

where $A_i,\ i = 1,2,3,4$ are given in the equation $(20)$ .

*2.6* ***Lnn-ELM algorithm for survival probability in the dependent renewal risk model (model 2) with Erlang(2) claim sizes***

Let us assume that the solution of (12) is

$$\Phi(u.) = \sum_{i=0}^{N} B_i.L_i(u) \qquad (22)$$

Putting (22) in (12) we have

$$\frac{c}{\lambda}\sum_{i=0}^{N} B_i L_i''(u) = \left(2+\frac{\gamma}{\lambda}\right)\sum_{i=0}^{N} B_i L_i'(u) - \frac{(\lambda+\gamma)}{c}\sum_{i=0}^{N} B_i L_i(u)$$
$$-\left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\sum_{i=0}^{N} B_i\int_0^u L_i(u-x)dF_1(x)$$
$$-\beta_i^2[1-F_1(u)]+\frac{\gamma\beta_2^2}{c}\sum_{i=0}^{N} B_i\int_0^u L_i(u-x)\ dF_2(x) \qquad (23)$$

The [0,b] can be divided into, $\Omega = \{0 = u_0 < u_1 < u_2 \ldots < u_M = b\}$

$$\frac{c}{\lambda}\sum_{i=0}^{N} B_i L_i''(u_j) - \left(2+\frac{\gamma}{\lambda}\right)\sum_{i=0}^{N} B_i L_i'(u_j) + \frac{(\lambda+\gamma)}{c}\sum_{i=0}^{N} B_i L_i(u_j)$$
$$+\left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\sum_{i=0}^{N} B_i\int_0^u L_i(u_j-x)dF_1(x)$$
$$-\frac{\gamma\beta_2^2}{c}\sum_{i=0}^{N} B_i\int_0^u L_i(u_j-x)\ dF_2(x),$$
$$=\beta_i^2[1-F_1(u_j)],\ \ forall\ \ j=0,1,2,3,\ldots M \qquad (24)$$

After arranging similar terms,

$$\sum_{i=0}^{N}\left(\frac{c}{\lambda}L_i''(u_j) - \left(2+\frac{\gamma}{\lambda}\right)L_i'(u_j) + \frac{(\lambda+\gamma)}{c}L_i(u_j)\right.$$
$$+\left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\int_0^u L_i(u_j-x)dF_1(x)$$
$$\left.-\frac{\gamma\beta_2^2}{c}\int_0^u L_i(u_j-x)\ dF_2(x)\right) B_i$$
$$=\beta_i^2[1-F_1(u_j)],\ \ forall\ \ j=0,1,2,3,\ldots M \qquad (25)$$

System of equations in (25) written in the matrix form

$$HB = Q \qquad (26)$$

where $H = \begin{bmatrix} LH \\ IH \end{bmatrix}_{(M+2)(N+1)}$, $IH = \{L_0(0), L_1(0), \ldots., L_N(0)\},$, $LH = \{lh_{ji}\}_{(M+1)(N+1)}$

$$lh_{ji} = \frac{c}{\lambda}L_i''(u_j) - \left(2+\frac{\gamma}{\lambda}\right)L_i'(u_j) + \frac{(\lambda+\gamma)}{c}L_i(u_j) + \left(\frac{\beta_1^2\lambda}{c}+\beta_1^3\right)\int_0^u L_i(u_j-x)dF_1(x)$$
$$-\frac{\gamma\beta_2^2}{c}\int_0^u L_i(u_j-.x)\ dF_2(x)\ \ forevery\ \ j=0,1,2,3,\ldots M$$

$$Q = \{\beta_1^2[1.-F_1(u_0)];\beta_1^2[1-F_1(u_1)];\ldots.;\beta_1^2[1-F_1(u_M)]\}_{(M+2)X1},$$

$$B = \begin{bmatrix} B_0 \\ B_1 \\ \cdot \\ \cdot \\ \cdot \\ B_N \end{bmatrix}_{NX1}$$

Hence [18] has a least square solution in the minimum-norm as $B = H^{\dagger}Q$, $H^{\dagger}$ is the generalized Inverse Moore-Penrose matrix of $H$ and lowest-norm among all the least square solutions.

### *2.7 MLPANN model development*

A Multilayer Perceptron Artificial Neural Network (MLPANN) model was developed for Model 2 using MATLAB machine learning application. It is a division of feed forward network [23][24]. In this model we used single input single output MLP network, as $u$ as input and $\Phi(u)$ as output. We have taken $100$ values of $\Phi(u)$ for training MLP model with corresponding $u$ values. Different values are used for testing. The below figure, Fig. 2 shows the MLP model. The algorithm used is Levenberg Marquardt. Fig. 3 shows R plot, Fig.4 represents mean square error plot.

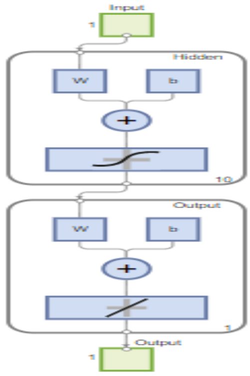


**Fig. 3**. MLP model

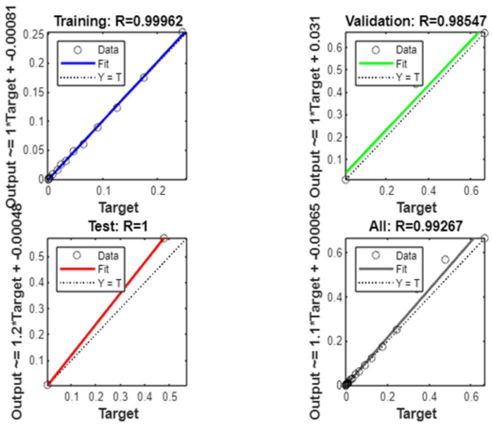


**Fig. 4**. R value for training and validatio

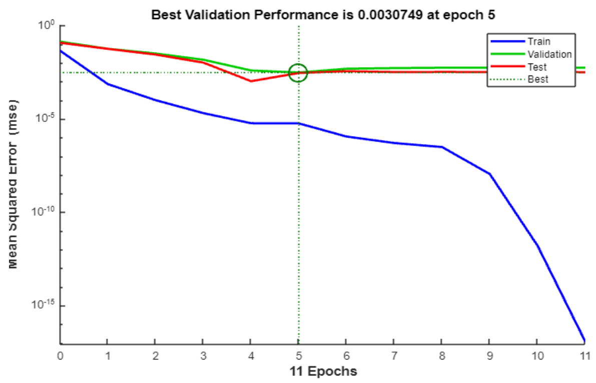


**Fig. 5**. Mean square error

## 3. Results and Discussions

In this session validation of the Model 2 is done using numerical applications. Acomparison of the Model 2 with Model 1 is also done with exact solutions. After that comparison between model 2, Lnn-ELM model and MLP models are carried out.

### 3.1 *Numerical applications*

Some numerical tests to find the solution of survival probabilities of Model 2 - Dependent Erlang(2) risk model is done here to demonstrate the supremacy of the developed Legendre orthogonal polynomial ANN approach. The traditional ELM algorithm uses randomly chosen parameters $w_i$ and $b_i$, where as in our Lnn-ELM algorithm we chose $w_i = 1$ and $b_i = 0$ The number of basis functions is governed by the validation set's execution duration and minimum mean squared error . $Mean\ Squared\ Error = \frac{1}{s}\sum_{i=0}^{s}\left[f(z_i) - \widehat{f(z_i)}\right]^2$

***Example:*** In the model 2 [15], when the claim size distribution is Erlang(2, $\beta$) with mean $\frac{2}{\beta}$.

For $x > 0$, $f(x) = \beta^2 x e^{-\beta x}$ The survival probability can be expressed as,

$$\Phi(u) = 1 + B_{s_0} A_2 e^{R_1 u} + B_{s_0} A_3 e^{R_2 u} + B_{s_0} A_4 e^{R_3 u} + B_{s_0} A_5 e^{R_4 u},$$

where $A_1, A_2, A_3, A_4, A_5$ are constants resulting from the limiting values of $s$ to $0, R_1, R_2, R_3$ and $R_4$ in $\chi(s)$ of (16) in [15].Training is done for the Lnn-ELM method with 20 equi-distant points of the stipulated interval [0, 10] with $b = 10$, $c = \lambda = \delta = \gamma = 1$, $\alpha = 2$, $\beta = 3$,M=20, $\beta_1 = 3$, $\beta_2 = 5$, $N = 12$ so that comparison can be done wisely

### 3.2 *Comparison of Model 2- dependent Erlang (2) model with Model 1- dependent Exponential model*

Table 1 shows a comparison of the Model 2- Erlang (2) risk model with Model 1- Exponential risk model. Also Fig.7 shows the comparison chart for values and corresponding variances. The variance for Erlang (2) model 2 is 0.0330 and that of the exponential model is 0.0433. The least variance of Model 2 shows its superiority over Model 1.

**Table 1.** Comparison between Model 2 and Model 1

| u | Survival Probability $\Phi(u)$ of Model 2 | Survival Probability $\Phi(u)$ of Model 1 |
|---|---|---|
| 0 | 0.3731121 | 0.3727340 |
| 0.5 | 0.3931121 | 0.3972734 |
| 1 | 0.4521231 | 0.4040040 |
| 1.5 | 0.4952032 | 0.4520485 |
| 2 | 0.5561101 | 0.5045698 |
| 2.5 | 0.5856121 | 0.5561890 |
| 3 | 0.6480231 | 0.6044159 |
| 3.5 | 0.6914851 | 0.6483483 |
| 4 | 0.7232051 | 0.6878524 |
| 4.5 | 0.7382301 | 0.7231337 |
| 5 | 0.7579822 | 0.7545300 |
| 5.5 | 0.7698108 | 0.7824151 |

|  |  |  |
|---|---|---|
| 6 | 0.7932913 | 0.8071559 |
| 6.5 | 0.8094290 | 0.8529095 |
| 7 | 0.8675661 | 0.8748543 |
| 7.5 | 0.8866781 | 0.8865781 |
| 8 | 0.8951236 | 0.8981058 |
| 8.5 | 0.9095021 | 0.9599969 |
| 9 | 0.9171722 | 0.9606595 |
| 9.5 | 0.9217543 | 0.9897227 |
| 10 | 0.9352126 | 0.9942665 |
| **Mean** | **0.719988** | **0.719608** |
| **Variance** | **0.033003134** | **0.043323** |

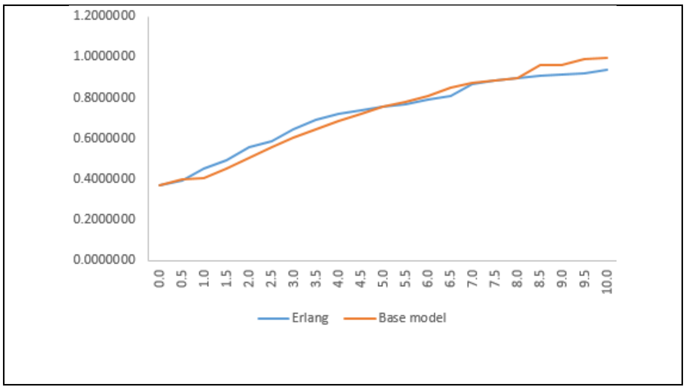


**Fig. 7.** Comparison of Erlang(2) model 2 and Exponential model 1

### 3.3 *Comparison of Model 2 with Lnn-ELM and MLPNN approach*

Table 2 shows comparison of the Model 2- Erlang (2) risk model with Lnn-ELM and MLPANN algorithms.

**Table 2.** Comparison between Model 2, Lnn-ELM and MLPANN

| u | Survival Probability $\Phi(u)$ of Model 2 Exact solution | Survival Probability $\Phi(u)$ of Model 2 with Lnn-ELM | Survival Probability $\Phi(u)$ of Model 2with MLPANN |
|---|---|---|---|
| 0.0 | 0.3731121 | 0.3918730 | 0.3730112 |
| 0.5 | 0.3931121 | 0.4591873 | 0.3973011 |
| 1.0 | 0.4521231 | 0.5949845 | 0.4821221 |
| 1.5 | 0.4952032 | 0.7059856 | 0.6056013 |
| 2.0 | 0.5561101 | 0.7316548 | 0.6848122 |
| 2.5 | 0.5856121 | 0.7723000 | 0.7531231 |
| 3.0 | 0.6480231 | 0.8107982 | 0.7821122 |
| 3.5 | 0.6914851 | 0.8202900 | 0.8294210 |
| 4.0 | 0.7232051 | 0.8313460 | 0.8346123 |

| | | | |
|---|---|---|---|
| 4.5 | 0.7382301 | 0.8405247 | 0.8469543 |
| 5.0 | 0.7579822 | 0.8492758 | 0.8581271 |
| 5.5 | 0.7698108 | 0.8341566 | 0.8656213 |
| 6.0 | 0.7932913 | 0.8456772 | 0.8772230 |
| 6.5 | 0.8094290 | 0.8467881 | 0.8881101 |
| 7.0 | 0.8675661 | 0.8578917 | 0.8891730 |
| 7.5 | 0.8866781 | 0.8689935 | 0.8993514 |
| 8.0 | 0.8951236 | 0.8791336 | 0.9033612 |
| 8.5 | 0.9095021 | 0.8892448 | 0.9104475 |
| 9.0 | 0.9171722 | 0.8993559 | 0.9195851 |
| 9.5 | 0.9217543 | 0.9038458 | 0.9219619 |
| 10.0 | 0.9352126 | 0.9148940 | 0.9295466 |
| Mean | 0.719988 | 0.78801 | 0.783409 |
| **Variance** | **0.033003** | **0.020161** | **0.030284** |
| **Coefficient of Variation** | **25.232 %** | **18.019 %** | **22.214 %** |

The variance for mathematical model developed is 0.0330 and that of Lnn-ELM and MLP model are 0.020161 and 0.030284 respectively. Here we can see Lnn-ELM model shows the least variance and hence the coefficient of variation also. The comparison chart is shown in Fig. 8.

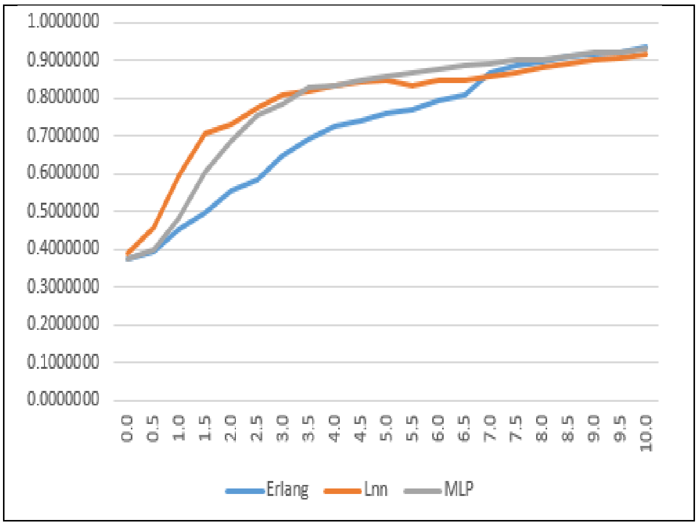


**Fig. 8.** Comparison of Erlang model 2, Lnn-ELM and MLPANN approach

## 4. Conclusion and Future Scope

In this paper, we considered two dependent insurance risk models – Erlang (2) risk model (Model 2) with Erlang (2) claim sizes and an Exponential risk model ( Model 1) with exponential claim sizes. First, we calculated the exact solutions and variance of both models and showed the superiority of the Erlang(2) model by comparing the variance with the exponential model (Table 1). Then we developed machine learning approaches of modern artificial intelligence - Legendre polynomial Neural Network- Extreme Learning Machine (Lnn-ELM) algorithm and Multilayer Perceptron Artificial NN (MLPANN) algorithm for the Erlang (2) model in MATLAB, compared with the exact solution (Table 2). We showed excellent agreement of the superiority and reliability of the Lnn-ELM method in finding survival probabilities as it is having very less variance compared to other machine learning method MLPANN and traditional method. Future study could concentrate on simplifying different risk model computations and expanding the Lnn-ELM model's applicability range.

## References


[1] H. Albrecher, O.J. Boxma, and J. Ivanovs, On simple ruin expressions in dependent Sparre Andersen risk models, *Journal of Applied Probability*(2014), *51*(1), 293-296.

[2] H. Albrecher and J.L. Teugels, Exponential behavior in the presence of dependence in risk theory, *Journal of Applied Probability* (2006), *43*(1), 257-273.

[3] F. Avram, A.L. Badescu, M.R. Pistorius and L. Rabehasaina, On a class of dependent Sparre Andersen risk models and a bailout application, *Insurance: Mathematics and Economics* (2016), *71*, 27-39.

[4] S. Asmussen and H. Albrecher, Ruin Probabilities, World Sci. *Publ. Co., Singapore*(2000).

[5] M. Boudreault, H. Cossette, D. Landriault and E. Marceau, On a risk model with dependence between interclaim arrivals and claim sizes, *Scandinavian Actuarial Journal*, 2006(5), 265-285.

[6] D.C. Dickson and S. Li, Finite-time ruin problems for the Erlang (2) risk model, *Insurance: Mathematics and Economics* (2010), *46*(1), 12-18.

[7] D.C. Dickson and C. Hipp, Ruin probabilities for Erlang (2) risk processes, *Insurance: Mathematics and Economics* (2010), *22*(3), 251-262.

[8] D.C. Dickson, B.D. Hughes and Z. Lianzeng, The density of the time to ruin for a Sparre Andersen process with Erlang arrivals and exponential claims, *Scandinavian Actuarial Journal*, (2005)(5), 358-376.

[9] T. Fujita, A. Sato, A. Narita, T. Sone, K. Iokawa, K. Tsuchiya and K. Otsuki, Use of a multilayer perceptron to create a prediction model for dressing independence in a small sample at a single facility. *Journal of physical therapy science* (2019), *31*(1), 69-74.

[10] H.U. Gerber and E.S. Shiu, On the time value of ruin. *North American Actuarial Journal* (1998), *2*(1), pp. 48-72.

[11] P. Gurevich and H. Stuke, Gradient conjugate priors and multi-layer neural networks, *Artificial Intelligence* (2020), *278*, 103184. S.Haberman, An Introduction to Mathematical Risk Theory, By Hans U. Gerber [SS Huebner Foundation, RD Irwin Inc. Homeward Illinois, 1979], *Journal of the Institute of Actuaries* (1980), *108*(1), 109-110.

[12] G.B. Huang, Q.Y. Zhu and C.K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* (2006), *70*(1-3), 489-501.

[13] D. Ibrahim, An overview of soft computing, *Procedia Computer Science* (2016), *102*, 34-38. P.S.G. Jagath, S. Naik and K.K. Suresh, Insurance Modelling under Natural Catastrophe for Dependent Risks and Survival Probabilities, *International Journal of Advanced Science and Technology*(2020), *29*(3), 3258 - 3272.

[14] R. Kaas, M. Goovaerts, J. Dhaene and M. Denuit, *Modern actuarial risk theory: using R* (Vol. 128), Springer Science & Business Media (2008).

[15] Y. Lu, G. Chen, Q. Yin, H. Sun and M. Hou, Solving the ruin probabilities of some risk models with Legendre neural network algorithm, *Digital Signal Processing* (2020), *99*, 102634.

[16] F. Lundberg, *1. Approximerad framställning af sannolikhetsfunktionen: 2. Återförsäkring af kollektivrisker. Akademisk afhandling*. Almqvist &Wiksells (1903).

[17] S. Mall and S. Chakraverty, Application of Legendre neural network for solving ordinary differential equations, *Applied Soft Computing (2016)*, *43*, 347-356.

[18] Q. Meng, X. Zhang and J. Guo, On a risk model with dependence between claim sizes and claim intervals, *Statistics & Probability Letters* (2008), *78*(13), 1727-1734.

[19] B.V. Norkin, The method of successive approximations for calculating the probability of bankruptcy of a risk process in a Markovian environment, *Cybernetics and Systems Analysis*(2004), *40*(6), 917-927.

[20] J.C. Patra and C. Bornand, Nonlinear dynamic system identification using Legendre neural network, In *The 2010 international joint conference on neural networks (IJCNN) 2010* pp. 1-7. IEEE. S. Radhakrishnan, S.G. Nair and J. Isaac, Multilayer perceptron neural network model development for mechanical ventilator parameters prediction by real time system learning, *Biomedical Signal Processing and Control* (2022), *71*, 103170.

[21] Y. Shirvany, M. Hayati and R. Moradian, Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations, *Applied Soft Computing (2009)*, *9*(1), 20-29.

[22] H. Sun, M. Hou, Y. Yang, T. Zhang, F. Weng and F. Han, Solving partial differential equation based on Bernstein neural network and extreme learning machine algorithm, *Neural Processing Letters* (2019), *50*(2), 1153-1172.

[23] K. Shakenov, Solution of Equation for Ruin Probability of Company for Some Risk Model by Monte Carlo Methods, In *Intelligent Mathematics II: Applied Mathematics and Approximation Theory (2016)* pp. 169-182, Springer, Cham.

[24] G.E. Willmot and J.K. Woo, *Surplus analysis of Sparre Andersen insurance risk processes*, Switzerland: Springer (2017).

[25] Y. Yang, M. Hou and J. Luo, A novel improved extreme learning machine algorithm in solving ordinary differential equations by Legendre neural network methods, *Advances in Difference Equations* (*2018)*, pp. 1-24.

[26] Y. Yao, V. Pérez-Rosas, M. Abouelenien and M. Burzo, MORSE: MultimOdal sentiment analysis for Real-life Settings, In *Proceedings of the 2020 International Conference on Multimodal Interaction (2020)*, pp. 387-396.

[27] T. Zhou, X. Liu, M. Hou and C. Liu, Numerical solution for ruin probability of continuous time model based on neural network algorithm, *Neurocomputing (2019)*, *331*, pp. 67-76.