International Journal of Future Generation Communication and Networking
Vol.14, No. 1, (2021), pp. 2977–2994

A Secure Aware SDLC with Hybrid Model of Artificial Neural Network-Firefly Optimization Algorithm for Estimating Security in Medical Software

Rayapati Venkata Sudhakar[1], Dr. B. Kranthi Kiran[2]

[1]Ph.D. ScholarJNTU Kukatpally, Hyderabad, Telangana, India,Email:rayapati1113@gmail.com.
[2] Associate Professor &Additional controller of Examination, Kukatpally, Hyderabad, Telangana, India, Email:kranthikiran9@gmail.com.

ABSTRACT:

The medical data requires an increase in use of security driven approaches to support software development activities, such as requirements, design and implementation. The main idea for practicing research towards security is to maintain usability of the software as well that was achieved by making less complex and high secure software. In order to meet our goal, we conducted a systematic mapping study to identify the primary studies on the use of software security techniques in Software Development Life Cycle (SDLC). The present research is intended to estimate the usable-security of software and achieves an objective of developing software with optimum security while retaining its usability. The decision-makers often find it difficult to integrate security and usability and therefore the present research hybrid Artificial Neural Network-Firefly Optimization Algorithm (ANN-FOA) work introduces an approach that integrates usability and security with its contributing attributes. The present research hybrid ANN-FOA significantly assess the usability along with security for SDLC model for medical field. The present hybrid ANN-FOA system introduces a procedural sensitivity was also achieved by using the various versions of the method. The findings of the usability along with security assessment insist that this inventive hybrid procedure would be the most conversant mechanism for determining the usable-security of software. Further these findings will be helpful in managing security without affecting the usability for end user.

Key words: *Hybrid Artificial Neural Network-Firefly Optimization Algorithm, Optimum security, Software Development Life Cycle*

1. Introduction

In this modern age, software-intensive systems have become an important part of our lives. We highly depend on software systems in several areas of daily activities, such as financial services, telecommunications, electronics, transportation, home appliances, and more. As the software system is involved in various aspects of society, security becomes an important issue and a vital requirement for the software system. The software product built should be of high quality that satisfies the client or end users from the organization choose the better Software Development Life Cycle (SDLC). The SDLC is varies from different organization from one another and therefore each of the organization has their self-policies and procedures and are different in terms

ISSN: 2233-7857 IJFGCN

of the infrastructure and the needs [1]. The software development projects have for getting the functionality in terms of time. The security is necessary and is relevant in terms of quality. The importance for cybersecurity has been considered widely over time and the quality of its consideration with the security is remained as an important technology nowadays. The traditional software development techniques related with security that considered cybersecurity nowadays. The traditional software development methodologies are introduced in the final phases and therefore the complex ones are increasing the possibilities generated additional costs, raising development times, and business reputation [2]. The quality has been taken for consideration but the security will be remained far and was considered but the importance for cybersecurity. The traditional software development methodologies security was usually introduced for final phases and therefore it would be most complex for increasing the additional costs, damaging business reputation, raising development [3]. Thus these features will provide the reasons for highlighted the importance implemented a Secure SDLC and the seemed to be imperative for the purposed of security was extended and was known that the secured SDLC. The concept of security was extended and was understood that security was needed to be present by the development [4].

The development of reliable software requires a software for the adoption the systematic processed a discipline addressed the security for each of the phases in the life cycle [5]. The types of security activities are integrated with the same stage: the first following secure design principles (minimum privilege, etc.) and the second including a series of security practices (specification of security requirements, cases of abuse, risk analysis, code analysis, dynamic penetration tests, etc.) [6]. This new life cycle with included security practices included is called S-SDLC. Among others, it is possible to mention some advantages of adopting an S-SDLC such as error identification or coding and design weaknesses in the early stages of development, which imply significant cost savings. Existing researches used the Software Development Life Cycle (SDLC) is a framework that defines the process used by organizations to develop an application from its origin to the end of its life cycle [7]. There are many software development methodologies and generally, all of them contemplate, from a high-level point of view, the following set of activities such as Identification of requirements Architecture and design Codification Testing Production and maintenance of the application. The vulnerabilities are assessed before the applications are deployed into the cloud. Various methodologies such as simulation process model, Multiple filters (FISHER and Maximum Response filters) and ANNIGMA wrapper approach etc., were developed for software security approaches for SDLC in cloud computing [8-10]. However, limitations like lack of clear description of context, more duration time, expensiveness of models gave rise to our proposed model. This research paper proposes a unified modeling language-based secure software maintenance procedure, where the proposed method is applied for maintaining a large-scale medical database. For secure designing of Web applications, this paper proposes system security performance model for trusted operating system. For re-engineering and re-implementation process of Web applications, the paper proposed the model-driven round-trip engineering approach.

ISSN: 2233-7857 IJFGCN

2. Literature Review

There are various number of approaches and tools available for incorporating the security approach in the SDLC. The main objective of the section is to determine the exisitng approaches used in SDLC thereby provides security firmly for developing security approaches for the new software that included various protection mechanisms such as firewalls, antivirus and intrusion detection systems. In the section, previously developed models are reviewed which are as follows:

M.I. Lunesu et al. [6] developed a simulation process that analysed various project parameters and the impact of parameters on projects in the cloud. This paper proposes a new software development process model that merges CBSE and MDD principles to facilitate software development. The developed model was tested successfully applied to the developed the e-learning system was exemplar for the case study. However, results showed that the simulations were stopped for specific cases that was served as an indication but failed to allow for the process of generalization. K. Fitzgerald et al., [7] developed an Agile Software Development that handled multiple file formats to generate professional and customized plots, without the usual steep learning curve. The developed model described how CMD Plot Tool was developed using Object Orientated Programming and a formal software design lifecycle (SDLC). The developed model required an astronomical software for developing the culture identified an appropriate paradigm for SDLCs. However, the developed Agile Software Development Lifecycle had lack of emphasis on necessary designing and documentation.

N.M Mohammed et al, [8] developed a systematic Mapping study that assisted an organization for the development for better understanding of software security techniques for SDLC. The main objective is to perform subjective decisions which was occurred at the time of paper selection and extraction of phases were used in the developed model. Due to lack of description in the objective or the context, the results failed to select based on primary studies. M.M Ali et al., [9] developed a Multiple Filters (FISHER and Maximum Response filters) and ANNIGMA wrapper based approaches. The information security is provided for the developed model that combined the filters and wrapper approach resulted better heuristic features for selecting the significant software metrics increased the performance in terms of accuracy when compared to the wrapper and independent filters. However, more filters used failed to analyze the combined parameters for performance improvement.

O. David et al., [10] developed an Object Modeling System Version 3 (OMS3) that was used for providing the features made the modelers easier for interoperable, contemporary, lightweight models, scalable by leveraging fully computing resources such as storing the data and to include the infrastructure opportunities. However, the model was not applicable in supporting the concepts in the code as there was no separation. Afrah Umran Alrubaee et al [11] developed a Process Model for Component-Based Model-Driven Software Development. The developed approach was pre-developed that used models throughout the process respectively. The developed model used a software development process that merged the MDD and CBSE principles facilitated

ISSN: 2233-7857 IJFGCN

the software development. However, the developed model that was based on interoperability was a challenge in CBSE failed to focus for defining the formal models and the MDD methods overcame the issues caused due to the interoperability.

Rajeev Kumar et al [12] developed a hybrid model that combined Fuzzy Decision Making Analysis estimated the usable security of the software. The usage of security of software was intended for estimating the usable security of software achieved in developing a software provided optimum security retained the usability. The developed hybrid model was used for integrating the security and usability found the decision makers. The developed approaches were used for integrating the usability and contributed the attributes for decision making to the security for software. However, the developed model which used hesitant fuzzy sets for decision making made the system computation complex. Katarzyna Filus et al [13] developed a Random Neural Network (RNN) which was used as a Bonding model for predicting the software vulnerability. The developed approach mixed the features by combining the text features for code generation using Static Code Analyzer. The developed RNN showed a bonding combined the text analysis and carried out the software using CNN and the outputs were generated using the Static Code Analysis. However, distinct static code analyses would have been used for identifying the features and was compared and evaluated for building the software vulnerability prediction tools.

Juan de Vicente Mohino et al [14] developed a Secure Software Development (S-SDLC) with Agile technologies. The developed model performed the interaction and integration which involved software life cycle and showed advantages showed immediate reaction to the change in reaction, implementing the artifacts or deliverables displayed various level progress was reached. The new software model used defines necessarily about the aspects used for providing the security in the SDLC and gets benefits using the Agile models. However, the developed model considered various scenarios for cloud environments that DevOps technologies overcomes the problems faced which addressed better the security issues and was useful. Marcus Sandberg et al [15] developed a Multidisciplinary Optimization of Life-Cycle Energy and Cost Using a BIM-Based Master Model that developed a framework for Building information modelling (BIM) neutral which is based on multidisciplinary optimization. The developed framework consisted of centralized master model that is having distinct specific domain based models were evaluated and generated. The developed model required a model expanded the account findings and energy was used for the process of construction.

Addressing security in the existing methods

Various approaches have been evolved for addressing the security in the software. The main approaches involved in addressing the software security are the patch secure operational and penetrate environment, secure operational environment and lastly secure software engineering. The penetrate and patch approaches released product was released once it was completed. If the vulnerability is found, then fixed and apply the patches. Applying the patches is the common approach for finding the vulnerabilities which would take up to 100 times and also expensive were

ISSN: 2233-7857 IJFGCN

fixed during the development. Securing the operational environment with more vulnerabilities was introduced when the patches were applied. The software systems relied on external devices provided protection and firewalls mechanisms. The external security is provided to the software and also gives security against the implementation attacks and the design. The operational environment security is achieved once after introducing the operational product. The main idea is to provide security for the well-structured process implemented with improved mechanisms from the software development phases. The secure software is initialized and is improved based upon the requirement face and is reflected on the entire stage of the SDLC.

3. Proposed method

There are two block diagrams are used in this research work. The one describes the software re-engineering and the other diagram describes the object-oriented class diagram for round-trip engineering.

Overview of SDLC

A reliable high-quality software modules are achieved by implementing the systematic and well-defined approach called SDLC, which describes the life cycle of software product. A quality of development process and software design are improved by SDLC techniques which is used to maintain, develop and replace the information systems. The developers such as software, QA, database, business analysts, end users and database designers are collaborated with SDLC for building the application software. The basic step in SDLC is to design the applications, fix the bugs occur in source code, documenting everything and then any improvements occurred in error-free code, that information is added to documents. The SDLC consists of six different phases such as requirement analysis and specification, design, coding, integration and testing, implementation and finally, maintenance. A preliminary investigation is the first step in SDLC, which is used to understand the needs and problems in present system. Whether the system can be able to develop or not is identified by the output of preliminary investigation.

The paper discusses a unified modeling language-based software maintenance process. In the proposed model, two major functions are performed. The first one is to develop a trusted operating system base secure engineering and second one is to analyze the model. In the proposed model, designing and developing of web applications. Firstly, the software will recognize which part is disabled for getting highest performance with respect to the web applications. Each and every security check constraints are worthy for all the software systems. Web applications can be described in unified modeling language with diverse object-oriented diagrams: Class diagram is used for application components; object diagram is used for object components; use case diagram is used for functional requirement and interactions with external systems. As we know, during software development process, it is very much common for software requirements to change and for faults to be corrected and removed. Each and every change in software may require that the

ISSN: 2233-7857 IJFGCN

UML object-oriented model be changed and a small change may lead to several other related changes. Consider a software system for supporting a large scale medical database.

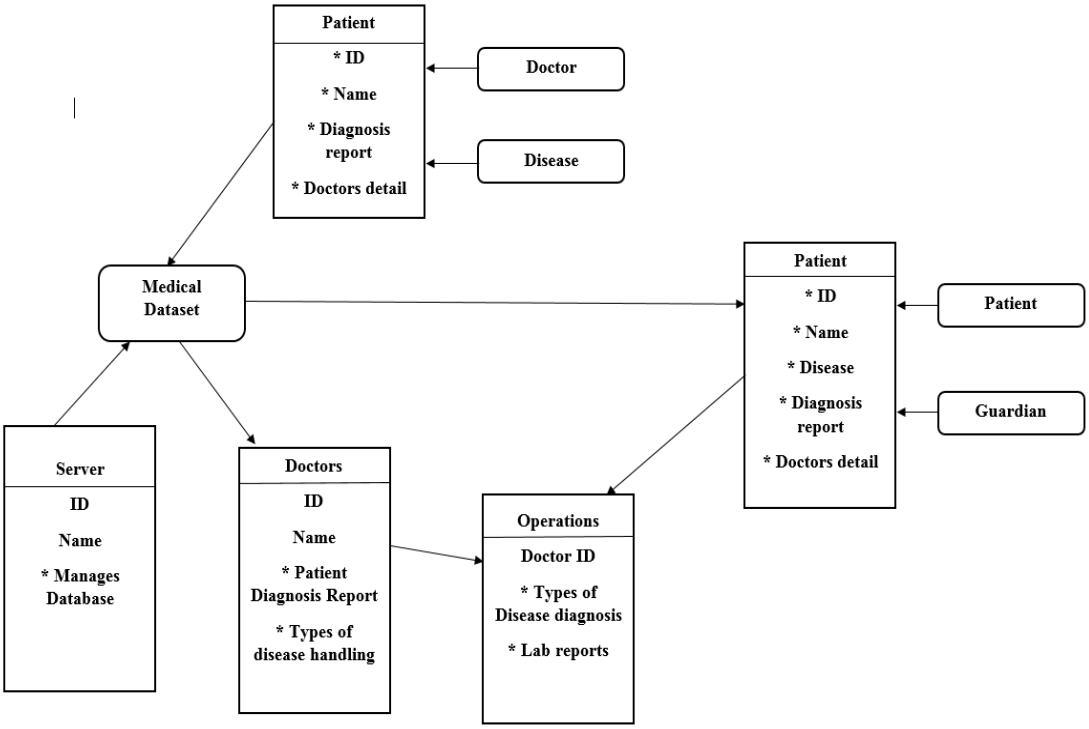

Figure 1: Object-oriented class diagram for round-trip engineering

Figure 1 indicates the object-oriented class diagram for medical data. The figure includes the information about the medical data of the patients as well other information related to doctor as well. The medical data here refers to the information such as Patient ID, Name, the type of disease, diagnosing the disease in detail. The similar categorical data information will be same in case of doctors as well. Doctor IDs, name, patient's diagnosis reports, operation information's, and the lab reports. The patient will be aware of the data related to the doctor they are getting treatment such as their area of specialty for disease diagnosis, name of the doctor and their ID as well. The security for such sensitive information is required in the present research work and thereby introduced an Artificial Neural Network model based SDLC model for medical data.

3.1 Data Collection

The model is a well-defined and rigorous method to identify, evaluate and interpret all the relevant studies regarding a particular research question, topic area or phenomenon of interest. Initially the data is collected that consists of Patient data and it is encrypted using Neural Network Cryptography, Neural Cryptography is the field of cryptography which analyzes encryption and cryptanalysis of artificial neural network. Encryption is the method of encoding the data or information which can be accessed only by authorized person. Cryptanalysis breaks the cryptographic security system and gets encrypted messages access.

ISSN: 2233-7857 IJFGCN

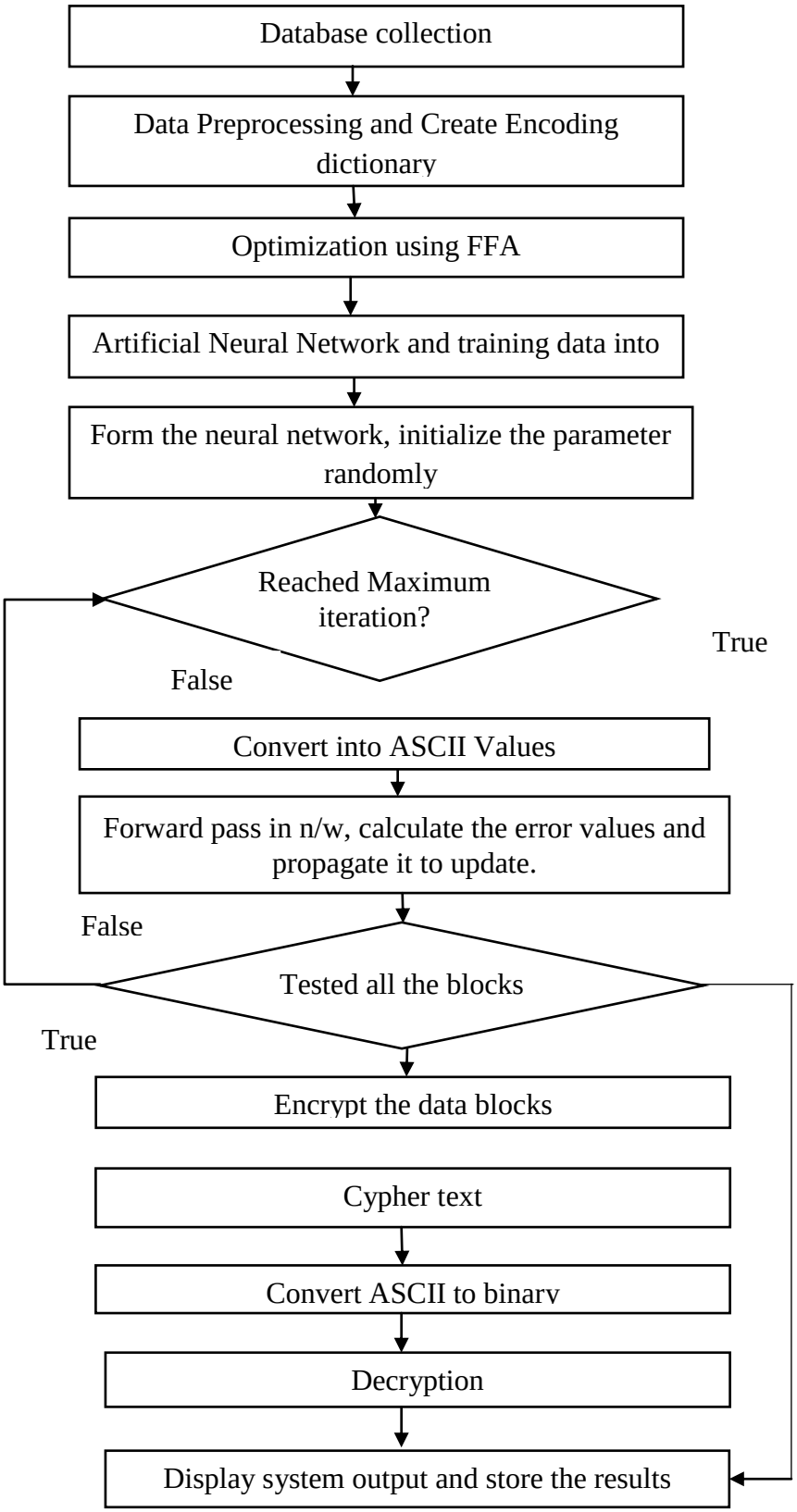

Figure 2 is related to general model for software re-engineering process for software development.

ISSN: 2233-7857 IJFGCN

3.2 Data Preprocessing

The data collected are now undergone for the process of pre-processing using Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering. Label Encoder encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to as assigned earlier. The categorical values have been converted into numeric values.

- Create an instance of Label Encoder () and store it in Label Encoder variable or object.
- Apply fit and transform which does the trick to assign numerical value to categorical value and the same is stored in new column called "State_N"
- Note that we have added a new column called "State_N" which contains numerical value associated to categorical value and still the column called State is present in the data frame. This column needs to be removed before we feed the final preprocess data to machine learning model to learn

3.3 Create dictionary Encoding

The pre-processing data is undergone for Dictionary encoding. Which is a type of data compression technique that can be applied to individual columns. Each of the column can be created with dictionary encoding in effect by applying the dict data handling property to the column during type creation using or create or type. An existing column can be converted to use dictionary encoding by modifying the column and applying the dict property (using or create or type).It will store each unique value of a column in memory and associate each record with its corresponding unique value. This eliminates the storage of duplicate values in a column, reducing the overall memory & disk space required to hold the data. An ordinal encoding involves mapping each unique label to an integer value.As such, it is sometimes referred to simply as an integer encoding.This type of encoding is really only appropriate if there is a known relationship between the categories.This relationship does exist for some of the variables in the dataset, and ideally, this should be harnessed when preparing the data.In this case, we will ignore any possible existing ordinal relationship and assume all variables are categorical. It can still be helpful to use an ordinal encoding, at least as a point of reference with other encoding schemes. The Ordinal Encoder () considers each variable for encoding it in to integers. This is a flexible class and does allow the order of the categories to be specified as arguments if any such order is known.The best practice when encoding variables is to fit the encoding on the training dataset, then apply it to the train and test datasets. From neural network, initialize the parameter randomly as it serves the process of symmetry-breaking and gives much better accuracy. In this method, the weights are initialized close to zero, but randomly. This helps in breaking symmetry and every neuron is no longer performing the same computation. Random initialization refers to the practice of using random numbers to initialize the weights of a machine learning model. Random initialization is one way of

ISSN: 2233-7857 IJFGCN

performing symmetry breaking, which is the act of preventing all of the weights in the machine learning model from being the same. In forward engineering, normally platform-independent object-oriented models are developed by software designers as part of software design document. In reverse engineering, these object-oriented models are usually derived automatically using model-driven transformations i.e. rethink, redesigning, specify and recode. To identify the resulting system, the functionality quality is compared with target system using optimization techniques using Firefly Algorithm (FFA). If the target system is achieved, then software re-engineering is formed.

3.4 Firefly algorithm for optimization

The pre-processing data is generated for each variable which will be encoded it in to integers. These integer values are driven automatically for selecting the best optimal solution from a huge data for SDLC model.The standard FA is very efficient, but there is still room for improvement. FA was proposed and is based on the flashing behavior of fireflies. Three basic rules are established for this algorithm.

1.The fireflies are unisex, and each firefly can be attracted to each other firefly.

2.The attractiveness and brightness are proportional, and their values decrease as their distance increases. For a couple of fireflies, the firefly with less brightness moves toward the other firefly; if they both have the same brightness, then their movement will be random.

3.The brightnessγ of a firefly is obtained by the objective function is shown in the Eq. (1).

$$\beta = \beta_0 e^{\gamma r^2} \tag{1}$$

The variation of attractiveness β with the distance r is defined by the following equation of attractiveness β with the distance r is defined by the following Eq. (2).

$$x_i^{t+1}{}_{=} x_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t \tag{2}$$

Where

x_i refers for the firefly position at the iteration t

$\beta_0 e^{-\gamma r_{ij}^2} \left(x_j^t - x_i^t\right)$ is defined as the attraction between the fireflies i and j

ϵ_i^t is the vector for random numbers generated using randomization which is defined as the α_t parameter.

The randomness is initialized base on the scaling factor is defined as using the Eq. (3)

$$\alpha_t = \alpha_t \delta^t$$

ISSN: 2233-7857 IJFGCN

Where δ is the value ranges from0 to 1. The values used for the present research work is applied as $\alpha, \beta,$ and δ. The obtained ranged values are trained into neural network blocks and therefore the present research uses ANN for initializing the random variables.

3.5 Encryption using ANN

The neural networks internally work with the binary data that consists of information such as (patient or doctor is male or female), numeric data, and categorical data that includes (such as hospital data, types of disease, diagnosis type the patient undergone etc) that should be encoded into numeric form. Additionally, the numeric data such as patient 'age, the date on which the person was undergone for treatment, weight etc. which are all numerical values are all normalized by the process of encoding. The data obtained are normalized and encode neural network data from a developer's point of view. The process is conceptually simple but surprisingly difficult to implement. The neural network in order to train the parameters obtained optimally brakes the raw data into individuals that are determined and stored as distinct binary and categorical values. The network layers' scan and tokenization the data for each iteration and computes the mean and standard deviation values that are the numeric data. If it is reached Maximum iteration then Convert the parameters obtained into ASCII values and Forward pass in network, calculate the error values and propagate it to update

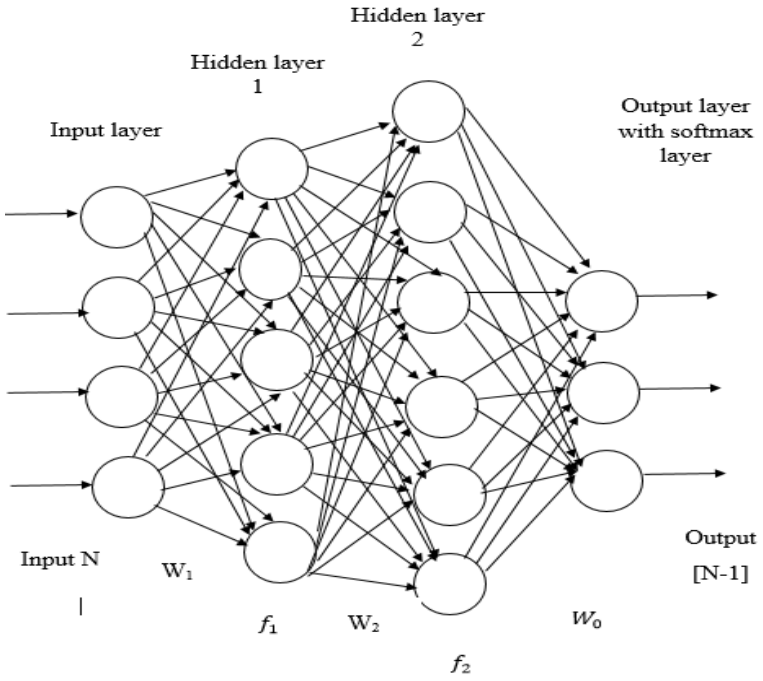

Figure 3: Structure of ANN

When training neural networks, forward and backward propagation depend on each other. In particular, for forward propagation, traverse the computational graph in the direction of dependencies and compute all the variables on its path. The networks are used for backpropagation

ISSN: 2233-7857 IJFGCN

where the compute order on the graph is reversed. Take the aforementioned simple network as an example to illustrate. The forward propagation is performed in the research that computes the intermediate variables in the output for the neural network from the input to the output layer which can be expressed using the below equation. In simple, an input sample where $\boldsymbol{x} \in \boldsymbol{R^d}$ and the hidden layer will not include the bias term Bwhich is calculated by an intermediate variable$z$ using the below equation :

$$z = W^{(1)}x$$

Where $W^{(1)}$ is is the weight parameter of the hidden layer.

The equation below shows the Net value $Net_h$for forward propagation for the Hidden layer H, B_h is the hidden layer with bias term, is given by using the below equation.

$$Net_h = W_{hx}^{'}X + B_h \,.H = \sigma(Net_h)$$

hidden activation vectorσhaving the Net_h value

$W_{hx}^{'}$Input layer to hidden layer

$$Net_o = W_{oh}^{'}H + B_o.\mathrm{O} = softmax(Net_o)$$

Hidden layer to output layer O is represented as $W_{oh}^{'}$

Error Function E

$$E = -\sum_{n=1}^{N}\sum_{k=1}^{250} t_{nk}\log(O_n k)$$

Error $E$for the training$t$ batched samples $k = 1to\ 250$

n is the number of iterations for these batch samples ranging from 1 to N, where N is a whole number

On the other hand, the gradient calculation for the parameter during backpropagation depends on the current value of the hidden variable, which is given by forward propagation. Therefore, when training neural networks, after model parameters are initialized, alternate forward propagation with backpropagation, updating model parameters using gradients given by backpropagation. The backpropagation reuses the stored intermediate values from forward propagation avoid duplicate calculations. The regularization term for forward propagation is depended upon the model parameters used currently. One of the consequences is that we need to retain the intermediate values until backpropagation is complete. This is also one of the reasons why training requires significantly more memory than plain prediction. Besides, the size of such intermediate values is roughly proportional to the number of network layers and the batch size. Thus, training deeper

ISSN: 2233-7857 IJFGCN

networks using larger batch sizes more easily leads to out of memory errors. To attain high quality throughout the secure Web application development, requirement discovery and analysis play an essential role. The objective of the research work is related with the understandability worth perceived by the user through code, i.e., also known as forward engineering process. If a software solution is being designed for the first time, our purpose is to be capable to properly model that software solution and to make as much of implementation/code from the object-oriented model. This will serve our motivation to enable IT services' companies to maintain object-oriented software development on several platforms. Our purpose is to reprocess as much of that software solution as possible in making that software solution accessible on several platforms. Tested all the blocks and Encrypt the data blocks thereby obtains the Cypher text. The cypher text is now converted from ASCII to binary values thereby decryption process is done. The system generates the outputs and are displayed and the results are stored.

4. Results and discussion.

The results of the proposed Artificial Neural Network based SDLC model results are simulated by Anaconda navigator and python 3.6 software with windows 10 operating system. The proposed model was implemented on system consisting of, 128 GB RAM, 1 TB memory configured with RTX 2080 Ti GPU and i9 processor operating at 3 GHz.

a. Performance metrics

The performance of the ANN-FOA is being assessed under different evaluation metrics such as its capability to classify the resources to the type of diagnosis made and its performance in secure retrieval in terms of time along with the encryption operation. The performance metrics are as follows.

Accuracy

Accuracy is defined as the measure of how much effective the ANN-FOA can classify the diagnosis operations validated from the resources given in other case it is defined as the measure of recognizing each diagnosis type correctly from the given pile of health records. This parameter serves mandatory to evaluate this parameter to classify the relevant resources together when it is in encrypted form.

$$\text{Accuracy } (\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\ \%$$

where, TP,TN,FP and FN represents True Positive, True Negative, False Positive and False Negative respectively

ISSN: 2233-7857 IJFGCN

Encryption Time

Encryption Time is denoted as the time required for ANN-FOA to encrypt a health record given as input to the system. This measure start from the time of reading the health record to the time instance where the encrypted data form is obtained.

Encryption Time = TimeInstance(CipherText) - TimeInstance(InputData) ms

Execution Time

Execution Time is a measure of time starting from reading the inputs given to the ANN-FOA model till the instance of time where the complete model is built and ready for further validation or ready for retrieval of health records.

Encryption Time = TimeInstance(RetrievalTime) - TimeInstance(InputData) ms

Mean Square Error

Mean Square Error is defined as the averaged of error occurred while classifying a single instance from the resource of health records validated in its squared form. MSE is necessary to evaluate the performance of ANN-FOA in retrieval of a wrong instance from dissimilar group of health records.

$$\mathrm{MSE} = \frac{1}{\mathrm{R}}\sum_{\mathrm{i}=0}^{\mathrm{R}}(\mathrm{Diagnosis}_{\mathrm{actual}} - \mathrm{Diagnosis}_{\mathrm{predicted}})^2$$

where, R is number of validated health records

$\mathrm{Diagnosis}_{\mathrm{actual}}$ is actual group of diagnosis for the corresponding patient data

$\mathrm{Diagnosis}_{\mathrm{predicted}}$ is predicted group of diagnosis for the corresponding patient data

Root Mean Square Error

Root Mean Square Error is defined as the standard deviation of the errors obtained in the prediction of diagnosis group in the validation patient data. RMSE analyses the probable deviation in the performance ANN-FOA classification and it gives an insight to minimize for the validation data.

$$\mathrm{RMSE} = \sqrt{(\frac{1}{\mathrm{R}} \times \sum_{\mathrm{i}=0}^{\mathrm{R}}(\mathrm{Diagnosis}_{\mathrm{actual}} - \mathrm{Diagnosis}_{\mathrm{predicted}})^2)}$$

b. Quantitative analysis

ANN-FOA model has the performance of classification accuracy about as 98.19 % data increases. The evaluation of proposed method in terms accuracy, precision and recall is being demonstrated in the figure below.Error and standard deviation of the prediction says that the proposed model

ISSN: 2233-7857 IJFGCN

doesn't have much deviation in terms of prediction and seems to be stable towards the increasing of data records. ANN FOA method proves to be the secure and optimal SDLC model for the retrieval of data with minimum error rates as shown in below figure 4.

Encryption time and execution time of the proposed model is found to be increasing minimum as the data get increases. And even the response for the query is minimal when compared to the existing methods. The model can able to perform validation for more than thousands of data within 1.5 seconds of time. That variation in time with respect to the number of data is being depicted in the below graph 5.

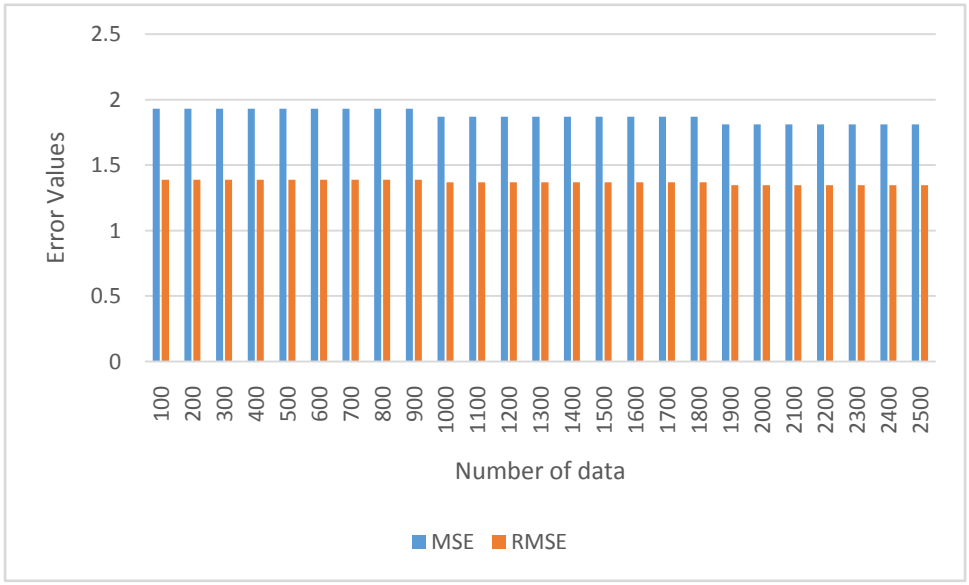

Figure 4 : Graph with respect to Error Values

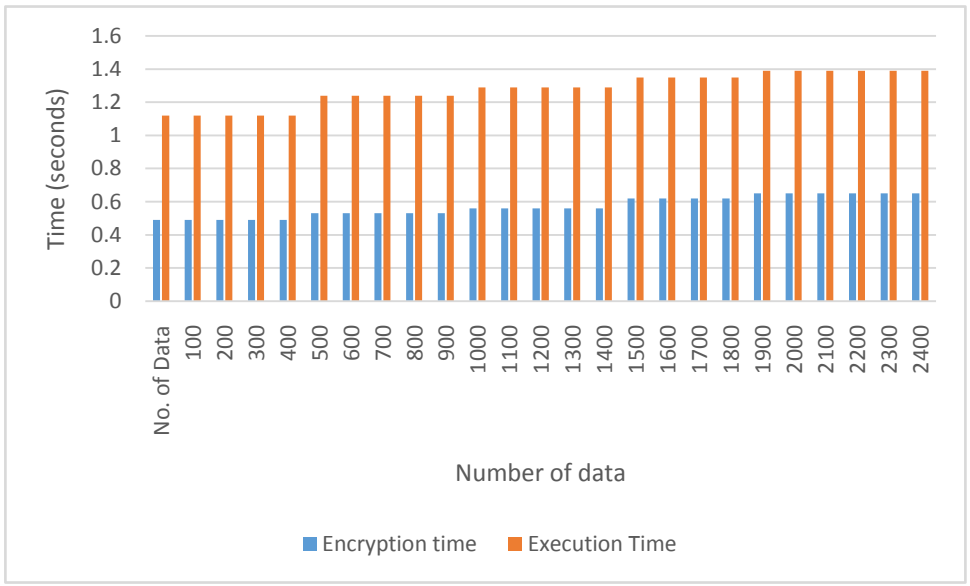

Figure 5 : Graph with respect to encryption time and execution time

ISSN: 2233-7857 IJFGCN

Table 1 : Comparison of the existing model with the proposed FOA-ANN method in terms of MSE

Number of data samples	**Optimization Algorithms**		
	LOA	**FOA**	**FOA-ANN**
100	3.48	2.42	1.93
500	3.37	2.40	1.93
1000	3.24	2.32	1.87
2000	3.19	2.29	1.81

Table 2 : Comparison of the existing model with the proposed FOA-ANN method in terms of Accuracy

Number of data samples	**Optimization Algorithms**		
	LOA	**FOA**	**FOA-ANN**
100	96.52	97.58	98.07
500	96.63	97.6	98.07
1000	96.76	97.68	98.13
2000	96.81	97.71	98.19

Table 1 shows the Comparison of the existing model with the proposed FOA-ANN method in terms of MSE and Table 2 shows the Comparison of the existing model with the proposed FOA-ANN method in terms of Accuracy. The existing FOA,at a specific location thebrightness of the firefly is selected asfor the problem of maximization. The is the relativeattractiveness that is determined by other fireflies as it varies with the distance. The intensity of light decreases as the distance between the source and light is absorbed by theair. However, the findings of the usability along with security assessment insist that this inventive hybrid procedure of combining the FOA with ANN would be the most conversant mechanism for determining the usable-security of software. The below figure shows the comparative analysis of the existing methods with the proposed method.

ISSN: 2233-7857 IJFGCN

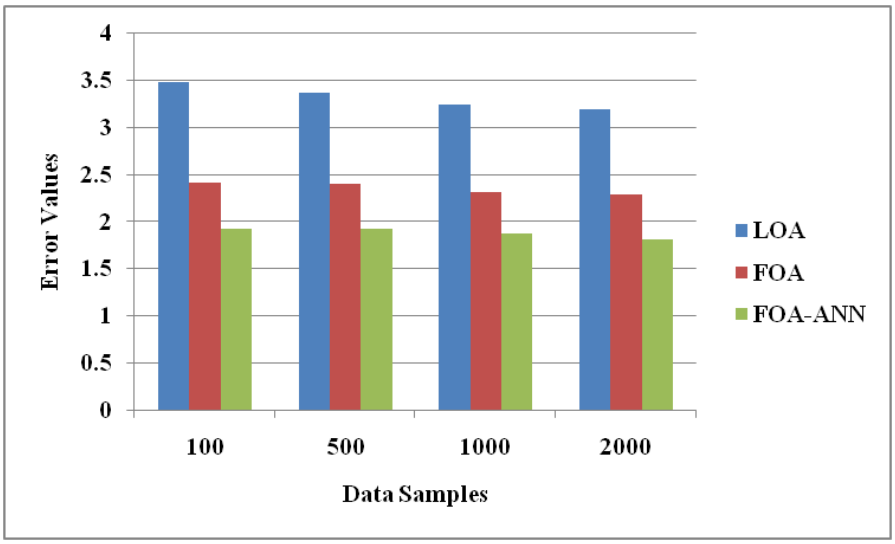

Figure 6 : Error values with respect to optimization algorithms

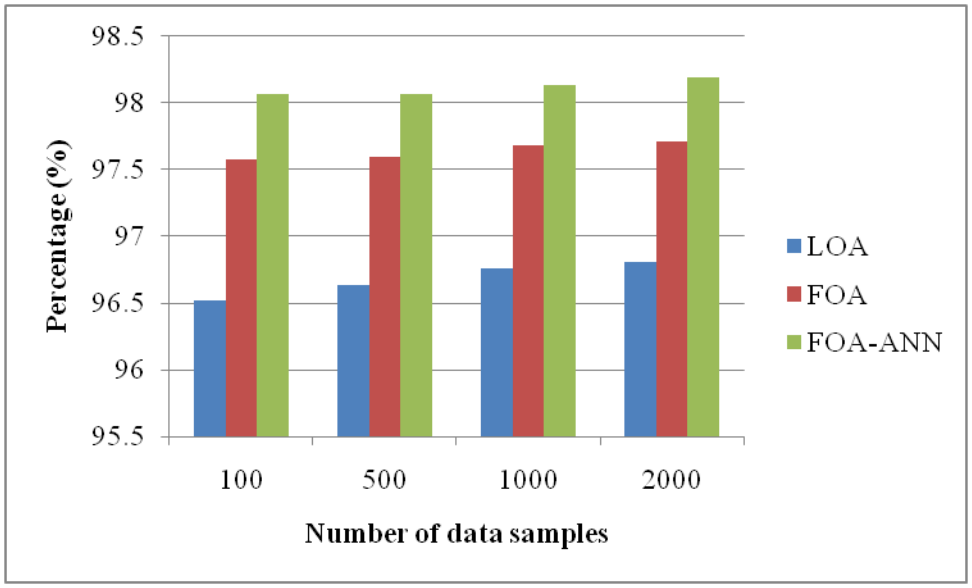

Figure 7: Accuracy values with respect to optimization algorithms

5. Conclusion

The main idea for practicing research towards security is to maintain usability of the software as well. This can be achieved by making less complex and high secure software. Though considerable efforts have been made in this context by the developers and security experts, the standard of usable-security is not the same as it should be. Therefore, it is important to research more on the most conversant mechanisms for assessing and increasing the usable-security of software. The objective of this paper is to identify the existing software security approaches used in the SDLC for medical data security. The current situation and

ISSN: 2233-7857 IJFGCN

discussion of this paper clearly defines that there is a need for usable as well as secure software at the same time in this digital era. The proposed hybrid ANN-FOA is facilitating effective usable- security into a software or software development process, all practitioners need a systematic path. An advantage of the proposed ANN-FOA is that it improved the security for the medical data as it performed assessment of software thereby supports the practitioners improves and suggest guidelines that further make software design more usable and secure. A framework which provides guidelines after testing the approach used might be proposed in the future which incorporates approach for assessment of usable-security.

REFERENCES

[1] Najjar, M., Figueiredo, K., Hammad, A.W. and Haddad, A., 2019. Integrated optimization with building information modeling and life cycle assessment for generating energy efficient buildings. *Applied Energy*, *250*, pp.1366-1382.
[2] Sharif, S.A. and Hammad, A., 2019. Simulation-based multi-objective optimization of institutional building renovation considering energy consumption, life-cycle cost and life-cycle assessment. *Journal of Building Engineering*, *21*, pp.429-445.
[3] Kaab, A., Sharifi, M., Mobli, H., Nabavi-Pelesaraei, A. and Chau, K.W., 2019. Use of optimization techniques for energy use efficiency and environmental life cycle assessment modification in sugarcane production. *Energy*, *181*, pp.1298-1320.
[4] Nabavi-Pelesaraei, A., Rafiee, S., Mohtasebi, S.S., Hosseinzadeh-Bandbafha, H. and Chau, K.W., 2017. Energy consumption enhancement and environmental life cycle assessment in paddy production using optimization techniques. *Journal of cleaner production*, *162*, pp.571-586.
[5] Paramesh, V., Arunachalam, V., Nikkhah, A., Das, B. and Ghnimi, S., 2018. Optimization of energy consumption and environmental impacts of arecanut production through coupled data envelopment analysis and life cycle assessment. *Journal of cleaner production*, *203*, pp.674-684.
[6] Lunesu, M.I., Münch, J., Marchesi, M. and Kuhrmann, M., 2018. Using simulation for understanding and reproducing distributed software development processes in the cloud. *Information and Software Technology*, *103*, pp.226-238.
[7] Fitzgerald, K., Browne, L.M. and Butler, R.F., 2019. Using the Agile software development lifecycle to develop a standalone application for generating colour magnitude diagrams. *Astronomy and Computing*, *28*, p.100283.
[8] Mohammed, N.M., Niazi, M., Alshayeb, M. and Mahmood, S., 2017. Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces*, *50*, pp.107-115.
[9] Ali, M.M., Huda, S., Abawajy, J., Alyahya, S., Al-Dossari, H. and Yearwood, J., 2017. A parallel framework for software defect detection and metric selection on cloud computing. *Cluster Computing*, *20*(3), pp.2267-2281.
[10] David, O., Ascough II, J.C., Lloyd, W., Green, T.R., Rojas, K.W., Leavesley, G.H. and Ahuja, L.R., 2013. A software engineering perspective on environmental modeling framework design: The Object Modeling System. *Environmental Modelling & Software*, *39*, pp.201-213.

ISSN: 2233-7857 IJFGCN

[11] Umran Alrubaee, A., Cetinkaya, D., Liebchen, G. and Dogan, H., 2020. A Process Model for Component-Based Model-Driven Software Development. *Information*, *11*(6), p.302.
[12] Kumar, R., Pandey, A.K., Baz, A., Alhakami, H., Alhakami, W., Agrawal, A. and Khan, R.A., 2020. Fuzzy-based symmetrical multi-criteria decision-making procedure for evaluating the impact of harmful factors of healthcare information security. *Symmetry*, *12*(4), p.664.
[13] Filus, K., Siavvas, M., Domanska, J. and Gelenbe, E., The Random Neural Network as a Bonding Model for Software Vulnerability Prediction⋆.
[14] de Vicente Mohino, J., Bermejo Higuera, J., Bermejo Higuera, J.R. and Sicilia Montalvo, J.A., 2019. The application of a new secure software development life cycle (S-SDLC) with agile methodologies. *Electronics*, *8*(11), p.1218.
[15] Sandberg, M., Mukkavaara, J., Shadram, F. and Olofsson, T., 2019. Multidisciplinary optimization of life-cycle energy and cost using a BIM-based master model. *Sustainability*, *11*(1), p.286.
[16] Ibrahim, A., Noshy, M., Ali, H.A. and Badawy, M., 2020. PAPSO: A power-aware VM placement technique based on particle swarm optimization. IEEE Access, 8, pp.81747-81764.
[17] Mosa, A. and Paton, N.W., 2016. Optimizing virtual machine placement for energy and SLA in clouds using utility functions. Journal of Cloud Computing, 5(1), pp.1-17.
[18] Qiu, H., Noura, H., Qiu, M., Ming, Z. and Memmi, G., 2019. A user-centric data protection method for cloud storage based on invertible DWT. IEEE Transactions on Cloud Computing.
[19] H. Qiu, H. Noura, M. Qiu, Z. Ming, and G. Memmi, "A user-centric data protection methodfor cloud storage based on invertible DWT". IEEE Transactions on Cloud Computing, 2019.
[20] G. Tian, H. Ma, Y. Xie, and Z. Liu, "Randomized deduplication with ownershipmanagement and data sharing in cloud storage". Journal of Information Security andApplications, vol. 51, pp. 102432, 2020.

ISSN: 2233-7857 IJFGCN