Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

RISK PREDICATION FOR SOFTWARE MANAGEMENT SYSTEM BASED ON MACHINE LEARNING

RABAB T. MAHMOOD
Department of Software, College of Computer and Mathematics, University of Mosul, Mosul, Iraq.
Email: ruba.altaee1986@gmail.com

IBRAHIM A. SALEH
Department of Software, College of Computer and Mathematics, University of Mosul, Mosul, Iraq.
Email: i.hadedi@uomosul.edu.iq

Abstract

Programming organizations strive to deliver excellent programming projects at a reasonable cost using the best available resources. Careful methodology must be followed to prevent risks leading to software failure to maintain software system level with global development of programming. The objectives of this study were to create and search risk prediction models using machine learning (ML) characterization methods such as Random Forest, Decision Tree, Logistic Regression and K-Nearest Neighbor. On the other hand, improvements were made to some algorithms using K-Fold and the gray wolf algorithm, which falls under the intelligence of the swarm, which showed a significant improvement in the mentioned algorithms. The experimental result of proposal methods indicates effectively predict with very high prediction accuracy.

Keywords: Machine Learning, Random Forest Software Risk, Risk Prediction, Swarm Intelligence, Logistic Regression

1. INTRODUCTION

Programming improvement faces inappropriate conditions that may negatively affect the outcome of programming development. Such events are called software risks [1]. These risks, in turn, have a negative impact on the software development process, which harms companies and programmers alike [2]. According to data by the Standish Group, just 16.2% of software projects are finished on schedule and on budget, while 31.1% of programs fail to reach their full potential [3]. The situation is considerably worse in large firms, where only 9% of programs are completed on time and within budget.

Only about 42% of the initially anticipated jobs were implemented in projects completed by major US companies [4]. There are two basic ways to approach risk forecasting: proactive and reactive. According to previous research, the absorption strategy is not a proven method for handling risk assessment because it reduces project quality and delivery while increasing cash schedule, and assets. To reduce the probability of important software being failed, a proactive methodology using AI procedures was implemented [5]. Additionally, risk prediction at this point is more advantageous and increases program development. Whenever risks are adequately managed throughout the needs collecting phase, it also aids in lowering the likelihood that a software project will fail [1].

This research attempts to solve the problem of risk prediction and its impact on the program's continuous production quality, schedule, and financial plan. Most current risk

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

prediction methods can use machine learning algorithms such as (k-nearest neighbors, Logistic Regression, Random Forest) to take a chance at later stages, usually from the product creation stage or product lifecycle code, in this way they may recognize problems but partially prevent them from occurring. [1]. Risks are caused by several triggers during Software Development Life Cycle (SDLC), which can be That, in turn, leads to the failure of the program project [2].

Making predictions of what was to come in light of verifiable and current information and her suggestions for pattern checking is the most common method of identification. Evaluation of the variable of interest at a particular future date is a continuous model. The broadest expression of determination is called prediction. The two terms can be used to depict formal strategies measurable using time series or longitudinal information, or they can be used similarly to depict less appropriate judgment actions. The expressions "predict" are sometimes used in the field of hydrology for projections of up-and-coming up-time values, although the expression "predict" is used for additional overall assessments of, for example, factors affecting something or the chance of a hazard or failure.

In light of the risk examination that includes preparation history and task attributes in the Product Improvement Life Cycle (SDLC), a product risk forecasting model was created. The model, known as the Atropos model, was accompanied by six main phases:[6].

- Gather information through the interface.
- Program similarities.
- Store date and time accounts.
- Similarity by setting accounts.
- Suggest any potential risks.
- Risk management and verification.

The recommended approach uses the identified necessity as input into determining the degree of risk associated with the prerequisites, allowing the undertaking supervisor or area manager to prepare and reduce risks only at an earlier stage.

The reset of paper Using Artificial intelligence to find associations between these variables, the objectives of the review were to identify key programming optimization factors that influence overall product task achievement or failure and to create risk prediction models that use AI clustering and search methods, for example, (KNN, LR, RF).

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

Figure 1: Atropos Six-Stage Model

2. RELATED WORK

A survey focusing on studies related to the risks of programming with artificial intelligence through different algorithms is presented in this section.

In [7] proposes a keen model that can recognize and control programming improvement gambles from a wide venture perspective. In this review, we fabricated a conventional model for risk distinguishing proof first, and afterward we accumulated true information from programming improvement firms to lay out a model for risk expectation. Fake Brain Organizations (ANN) and Backing Vector Machine (SVM), two AI calculations, are differentiated to survey the exhibition of our model. The consequences of the examinations show that our SVM-based risk expectation model performs all the more precisely while making forecasts.

In [8] introduce computer methodology is suggested for lowering project failure probabilities through risk forecasting. The study's goal is to show how a model may assist teams in identifying and managing risks at various stages of a project's lifespan. As historical contextual listings, 153 software projects from a financial organization were taken from a dataset. Results: The proposals were examined by a project team of 18 specialists, who scored them 73% acceptably and 83% accurately when compared to initiatives that had already been carried out. Comparing the results to other models that do not take into account the traits and similarities of projects, they revealed a high percentage of risk suggestion acceptance.

In [9] there was a work to forecast the time, money and asset risks facing transferred groups in light of improving worldwide programming. Brain network approaches such as

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

Levenberg-Marquardt, Bayesian Regularization, and Scaled Form Slope to predict risk interactions associated with project time, cost, and assets associated with programming progress were implemented worldwide to examine the overall impact of these variables. For the most noteworthy accurate account, the correlation investigation of these three accounts is also completed. The findings of this study showed that, in contrast to the Levenberg-Marquardt and Scaled Form Slope strategies, Bayesian regulation was characterized on the basis of MSE (consent).

In [10] few exploration have utilized various hypotheses and philosophies to figure programming issues. Fake brain organizations (ANN), arbitrary backwoods (RF), irregular trees (RT), choice tables (DT), direct relapse (LR), Gaussian cycles (GP), SMOreg, and M5P are utilized in this request. To figure impending programming blemishes, an original forecast model is put out. The figure of deformities depends on past data. The discoveries show the way that an assortment of ML calculations can be utilized to precisely figure programming deficiencies. The SMOreg classifier played out the best, while the ANN classifier played out the most unfortunate.

In [11] recommended a new system for the information got from the poll, where risk evaluations for every one of the distinguished programming models were completed utilizing AI classifiers. Programming item chiefs can pick the best programming model in light of the product prerequisites and hazard expectation rate utilizing this outcome.

In [10] the researchers created ML models to predict defects and risks in the programming execution space. ANNs, Random Forest (RF), Random Tree (RT), Decision Tree (DT), Logistic Regression (LR), Models were made using the accompanying strategies: ANN, irregular forests, random tree, selection table, direct regression, Gaussian cycles, SMOreg, and M5P. The NASA Commitment Store provided the information to re-enact these actions. The results show that the combination of different ML calculations is effective in predicting programming errors. Relationship coefficient (R2), mean explicit error (MAE), root relative squared error (RMSE), relative explicit error (RAE), and relative explicit error (RAE) were the evaluation networks used (RRSE). Limitation and future exploration: Results can be surveyed using a variety of datasets and machine learning strategies. To upgrade these results, additional examination of programming perspectives must be completed.

In [12] refined the beneficial expenditure (COCOMO-II) model in structure of GSD to cheaper risk. The review was directed, and 175 responses were obtained, resulting in a diverse set of data. Scale proportional evaluations (MRE) and skillful judgment were used to evaluate the model's measurements. Barriers and future work: On the basis that the model is still under development, further approval and evaluation is needed. Later, it may also be possible to use numerical or artificial intelligence (ML) procedures.

In [13], A model was created using ANN to measure risk factors for GSD. Levenberg-Marquardt procedures, Bayesian organization, and Scaled Form Slope were used in the model. 760 surveys outside the organizations were shipped to compile the data set. 274 out of 390 responses were used as main information indicator after excluding 116 responses. The mean least square error (MSE) was used to determine the model

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

evaluation measurements, and the results showed that Bayesian regularization provided better results than the other two methodologies and matched the findings of these two examinations [14]. The example informational index needs to contain a few organizations, and irregular information assortment ought to be used to sum up the model, which is one of the model's deficiencies and regions that necessities further exploration. To get future experiences and discoveries that are more precise, the creator likewise recommended taking on profound learning.

3. METHODOLOGY

This section describes the methodology used to develop the software risk prediction model shown in Figure(3), the five stages of the proposed model. Data set selection, data set pre-processing, modeling, experimental evaluation, and risk prediction results.

3.1 Dataset Selection

The dataset used conation (12) Feature they are requirements, project, requirement, risk. Probability, Magnitude, Impact, Dimension, Affecting, Fixing, Fix, and Priority. The attribute function is “Risk. Table (1) shows The Features of dataset used in this work. The characteristics of a software project's risks and requirements make up the risk profile. The dataset used in this work was taken from the Expected Programming Requirements Hazards dataset located at https://doi.org/10.5281/zenodo.1209601. 299 cases of risk prediction information were submitted in the information indicator during the initial phase of the product improvement life cycle (prerequisite gathering phase). Twelve (12) quality factors and one (1) objective variable make up the data set (outcome). Because the characterization task predicts discrete reactions, it is encouraged and applied to information that can be ordered, welcomed, or isolated into unambiguous groupings or categories. The probabilistic, hazard magnitude, effect, need, and adventure properties have been changed along with the levels to reflect the direct attributes that can be used by the different classifiers. The target variable is the level of risk.

Table 1: Dataset Features

No	Feature Title
1	Requirements
2	project
3	Requirement
4	Risk
5	Probability
6	Magnitude
7	Impact
8	Dimension
9	Afftecting
10	Fixing
11	Fix
12	Priority

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

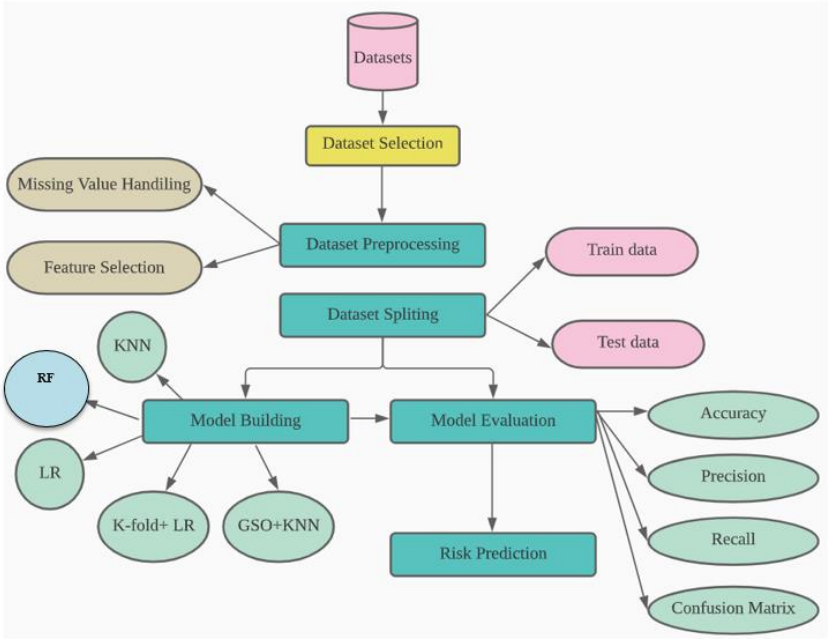

Figure 2: Proposed system

3.2 Dataset Preprocessing

The first stage for information pretreatment could be viewed as the fundamental phase of making an AI model. True data is habitually lacking, conflicting, or off base (since they have exceptions or mix-ups). Preprocessing techniques should be utilized to rebuild the dataset to keep it clean, organized, and organized [15]. The dataset preprocessing techniques in this subsection are dataset distinguishing for the following:

- **Find and manage missing features:** False detections may be generated due to insufficient information. In this way, these conditions can be dealt with by defining the means of the properties such as (Requirements, project, Requirement, Risk, Probability, Magnitude, Impact, Dimension, Afftecting, Fixing, FixPriority) using mathematical information. This is more applicable than the common methods of taking care of missing attributes, which involve leaving out the line or passage entirely in light of the fact that doing so may cause the data set to be one-sided or to mistakenly take out information. Alternative terms include median, mean, and formula.
- **Coding straight out information:** AI, as is notable, just works with mathematical characteristics. Thus, absolute credits are pointless until they are changed over into mathematical data. In this way, only four unmitigated properties ought to be changed over into mathematical qualities: area, work, progress, and procedure. The class ascribes were changed over into mathematical qualities utilizing the name encoder approach of the Python scikit-learn module. As indicated by this strategy, an unmistakable whole number is given to each mark in view of the sequential request [16].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

- **Feature selection:** Feature extraction using correlation is a technique used in data analysis and machine learning to identify and extract relevant features from a dataset. The basic idea is to calculate the correlation between each feature in the dataset and the target variable and select the features that have the highest correlation with the target. Correlation is a statistical measure that describes the relationship between two variables. In the context of feature extraction, the target variable is the variable we are trying to predict, and the features are the variables that we use to make the prediction. To extract features using correlation, we follow these steps:
  - Calculate the correlation between each feature and the target variable. This can be done using Pearson's correlation coefficient, which measures the linear relationship between two variables.
  - Rank the features based on their correlation with the target variable. Features that have a high correlation with the target variable are more likely to be relevant for prediction.
  - Select the top N features with the highest correlation and use them for prediction. The value of N depends on the size of the dataset and the complexity of the prediction problem.

Feature extraction using correlation can be a useful technique for reducing the dimensionality of a dataset and identifying the most important features for prediction. However, it is important to note that correlation does not always imply causation, and other factors may also be important for prediction. Therefore, it is important to combine feature extraction using correlation with other techniques, such as feature selection and regularization, to ensure that the final set of features is optimal for the prediction problem. Independent items of interest should be selected using correlation. Accordingly, unimportant elements or attributes of the form should be excluded. The Connectivity Examination, a multivariate investigation (EDA) in light of measurable strategies, was performed to discover these connections to assess the direct correlation between properties and each other and to gain a better understanding of the parts and their connections [26]. The thickness between two factors is estimated involving the relationship coefficient as the unit of estimation. The last order model is irrelevant to the predetermined connection correlation-based feature selection (CFS) strategy in this work since it is a sifting technique. Just the natural characteristics of the information — connections, as the name as of now suggests are utilized to assess include sets. The goal is to distinguish a subset of qualities with low element connection to forestall reiteration and high component class relationship to safeguard or work on prescient power.

To do as such, the calculation utilizes the accompanying condition to gauge the value of a subset s with k Features:

$$Merits = \frac{k_{\overline{rcf}}}{\sqrt{k+k(k-1)\overline{rff}}} \quad \text{………………………………… (1)}$$

$\overline{rff}: average\ feature - feature\ correlation$

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

$\overline{rrf}$: average feature − class correlation

$\bar{k}$: number of features of that subset

Accordingly, include connection can be determined utilizing the normal Pearson relationship coefficient. The component class connection is in any case resolved utilizing the two-point relationship coefficient on the grounds that the informational index's class is paired.

3.3 Machine Learning

Various calculations are used by AI (ML) to analyze the input information and predict the output. The labeling system is automated with manual calculations of artificial intelligence, which additionally learns and improves execution. Each ML calculation is made to complete a specific capacity. Three AI (ML) calculations were tested in this review for their ability to measure programming issues, and two of them were upgraded using K-Fold and swarm intelligence. Here is a summary of these ML algorithms: [27].

3.3.1 Random Forest (RF)

Leo Breiman created Random Forest, a variety of irregular trees or regression made from random scanning of preparation information. Features are selected during the extrapolation technique. Costume expectations were included (the bulk voted in favor of the characterization, normal for relapse). Each tree is generated as per[16].

- In the event that the preparation set contains N cases, yet with substitution, by haphazardly choosing N cases. The preparation set for developing the tree will be this example.
- For M number of information factors, the variable m is chosen with the end goal that m<<M is indicated at every hub, m factors are chosen aimlessly out of the M and the best parted on these m is utilized for parting the hub. During the Forest developing, the worth of m is held steady.
- Each tree is developed to the biggest conceivable degree. No pruning is utilized.

The steps of evaluation Random Forest Algorithm can be summarized in three steps, the first one is to build a random vector, and use this random vector to build multiple decision trees and combine these trees. Figure (3) describe the RF algorithm.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

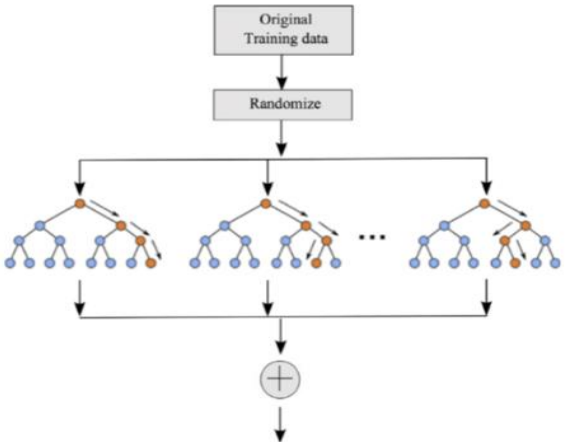

Figure 3: Random Forest Algorithm steps

3.3.2 K-Neatest Neighbor (KNN)

A relationship based ML calculation is the KNN. As a general rule, it decides the division between focuses utilizing the Euclidean distance and afterward names new information in view of the marks of the close by data of interest [5]. The equation for the Euclidean distance is:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \text{..... (1)}$$

where (x,y) is the two-point Euclidean N-space, $(x_i - y_i)$ addresses the Euclidean vectors, and d is the Euclidean distance (attribute numbers).

In postulations steps, the K-NN working can be made sense of.

- Decide on the neighbors' K-numbers.
- Calculate the Euclidean distance between K neighbors in step two.
- Based on the calculated Euclidean distance, select the K neighbors that are closest to you.
- Count the amount of the relevant data for each class among these k neighbors.
- Distribute the new information to the classification for which the neighbor quantity is the highest.

A populace based metaheuristic can be integrated into the GWO algorithmic structure to view as the ideal nearby ideal. Cosmic movement influences the development conditions in GWO [17]. GWO isolates the populace into subpopulations, every one of what capabilities as a sub swarm [17]. The worldwide outperforms are assembled to make the super swarm, which additionally takes part in PSO development, and the sub swarms are dependent upon PSO development [18].

GWO is an algorithmic system that empowers non-interfacing metaheuristic sub swarms to independently investigate the inquiry space. Investigation is a critical part while

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

handling multimodal issues. In GWO, the sub swarms don't convey. The super swarm ought to have solid neighborhood search capacities, and the sub swarms ought to be forceful in their overall pursuit. When contrasted with a solitary populace based strategy, substituting sub swarm and super swarm developments over various ages takes into consideration a rotation of worldwide and neighborhood search, creating improved results [18].

3.3.3 Logistic Regression (LR)

A grouping strategy in view of factual procedures is the calculated relapse calculation. A well-known AI (ML) approach for twofold characterization that puts together expectations with respect to the sigmoid capability is strategic relapse [19]. Especially suitable for direct regression model, strategic regression is a characterization technique that predicts dual factors. The variable Y in strategic regression has two possible attributes: 0 (solid line) and 1. (Broken pipe). In 1969, Dr. Cox was quick to make a strategic regression model, which is currently generally used in both social and innate science tests. The relevance of this form includes:

- Describe the relationship between the continuous variable (the independent variable) and the probability of disappointment (the dependent variable). F is a double variable, defying the typical publishing law.
- Need Executives (Software Fail)
- Formation of the indicator model. To predict the limits using the most extreme scaling approach, the model does not use the least squares technique.

K-Fold was used in the ongoing review to evaluate recommended models. Figure (4) shows a flowchart for running the plant cross-validation process. The K-Fold cross-validation procedure is described as follows:

- Data is divided into K folds.
- K-1 sets are taken from the K folds, and the remaining set is used for testing.
- The model is then prepared and tried K times, with each test involving an alternate set for preparing and the excess sets for testing.
- The normal of the results from each set comprises the K-Fold cross-validation result.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

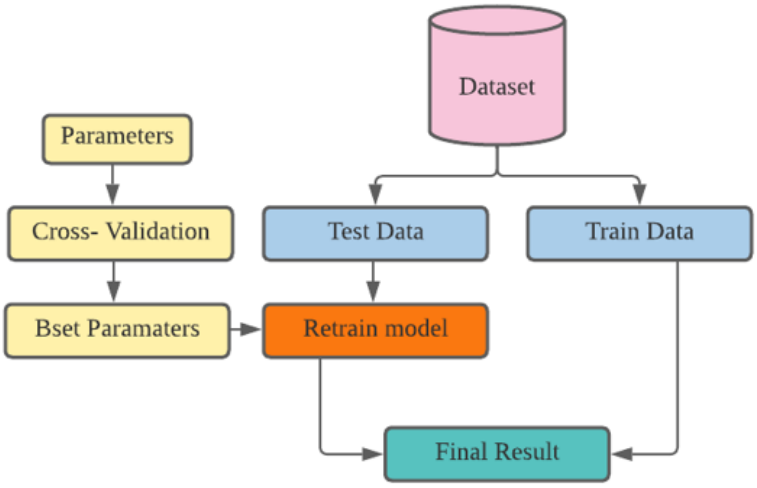

Figure 4: K-fold Evaluation Model

3.4 Evaluation Metrics

When training a classifier, the assessment scale is crucial to getting the best classifier possible. So selecting the right rating scale is crucial for differentiating and getting the best classifier. In order to enhance the generative classifier, this section has thoroughly analyzed pertinent assessment metrics that are intended to act as discriminators. Generally speaking, precision is a measure that many generative classifiers employ to identify the best solution while training [20]. Accuracy has various drawbacks, including being less informative, less discriminatory, and biased against data from the dominant class. Other measurements that are explicitly intended to define the ideal solution are briefly included in this paragraph as well. Also mentioned are these alternate measures' drawbacks [20].

Confusion matrix for a binary classifier (Figure 5). Actual values are marked true (1) and False (0), and are predicted as Positive (1) and Negative (0). Estimates of the possibilities of classification models are derived from the expressions TP, TN, FP, FN, which exist in the confusion matrix [21].

Class designation		Actual class	
		True (1)	False (0)
Predicted class	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 5: Binary Classification Confusion Matrix [20]

- **TP (True Positive)** - The data of interest in the disarray framework is the genuine positive point (TP) when a positive result is normal and exactly the same thing occurs.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

- **FP (False positive)** - The data of interest in the disarray lattice is a bogus positive when a positive outcome was normal, and what happened is an adverse outcome. This situation is known as a sort 1 mistake. It resembles the gift of awful premonition.
- **FN (False Negative)** - The data of interest in the disarray lattice is bogus negative when an adverse result was normal, and what happened is a positive result. This situation is notable as a kind 2 blunder and is considered as perilous as a sort 1 mistake.
- **TN (True Negative)** - The data of interest in the disarray framework is Valid Negative (TN) when an adverse result is normal and the equivalent occurs.

Table 2: The elements of the evaluation process (variables, definitions, and equations) [22]**.**

Variable	Definition	Equation
Accuracy	the percentage of accurately anticipated data from tests is easily determined by dividing all accurate forecasts by all predictions.	$Accuracy = \frac{Tp+Tn}{TP+TN+FP+FN}$
Precision	he proportion of outstanding instances among all anticipated ones from a specific class	$Precision = \frac{TP}{TP+FP}$
Recall	the ratio of the total number of occurrences to the proportion of instances that were supposed to be members of a class	$Recall = \frac{TP}{TP+FN}$
F1-Score	The phrase is used to describe a test's accuracy. The maximum F1-score is 1, which denotes outstanding recall and precision, while the lowest F1-score is 0.	$F1 - Score = 2 \times \frac{percision \times recall}{Percison+recall}$

4. RESULTS

Machine learning algorithms were applied according to the above-mentioned scheme, where the implementation started using the (RF) algorithm, which gave an accuracy of its value (97.78%). Then the system was trained on the (LR) algorithm, which gave a low accuracy of (62.22%). Then the (KNN) algorithm was used, which gave a low accuracy of (88.89%). After that, an improvement was made on the algorithms that gave the doubling, where the K-Fold classifier was used with the (KNN) algorithm, which increased the accuracy, reaching (93%). The intelligence of the swarm represented by the (GWO) algorithm was used, as the accuracy increased to (96.76%). The figures below show the confusion matrix for each of the above mentioned algorithms.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

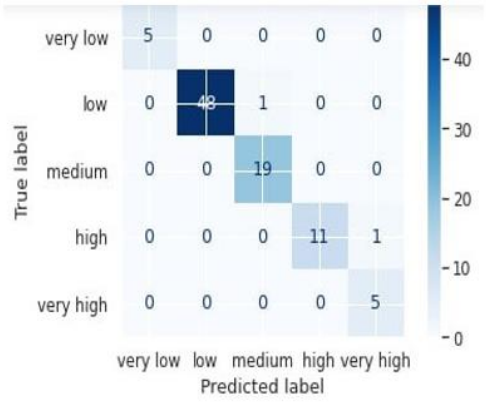

Figure 6: RF confusion Matrix

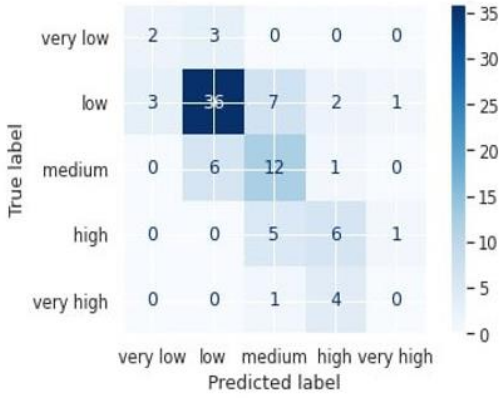

Figure 7: LR confusion Matrix

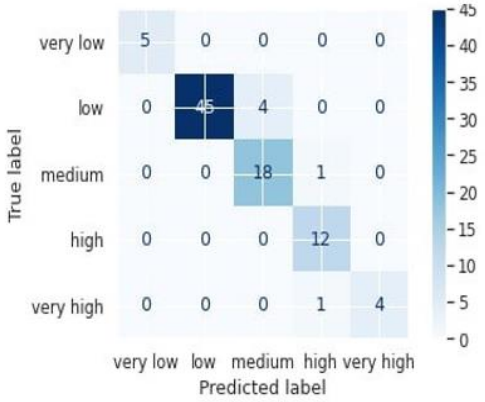

Figure 7: KNN confusion Matrix

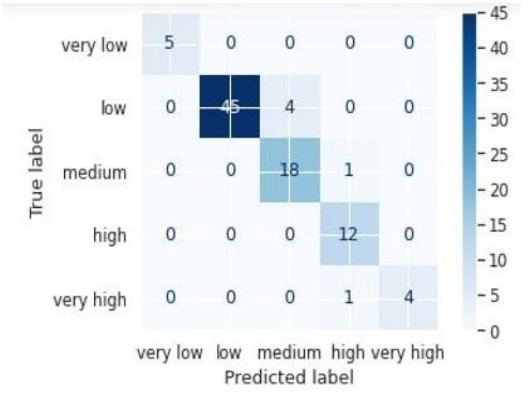

Figure 8: K-Fold+ LR confusion Matrix

The classification results showed that Random Forest gives better results for the same number of features and large data sets, K-Fold and GWO algorithms also contributed to the improvement of machine learning algorithms, as the accuracy of an algorithm increased KNN by (8%), and LR by (31%). Also, the results compare to relative work results as shown in Table (4).

Table 3: Algorithms Results Comparison

Algorithm	Accuracy	Precision	Recall	F1-Score
RF	98%	96%	98%	97%
LR	62%	43%	45%	445
KNN	88%	94%	93%	93%
K-Fold+ LR	93%	94%	93%	93%
GWO+ KNN	96.67%	-	-	-

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

Table 4: Relative work comparison

Researcher	Dataset	Algorithm	Metric	Accuracy
Yong et. al.,	Collected manually	ANN	Accuracy	73.0%
Sahu & Srivastava	From John Musa of bell laboratories	NN	Accuracy	54%
Assim et. al.,	From NASA Promise	RF	Accuracy	73.19%
		LR	Accuracy	75.94%
KHAN et. al.,	https://data.mendeley.com/datasets/m2zr2ns5k7/1	ANN	Accuracy	75%
Iftikhar et. al.,	-	ANN	Accuracy	78%
Filippetto et.al.,	-	NLP	Accuracy	83.00%
Assim et. al.,	-	ANN	RMSE	14.7296
		RF		9.3905
		RT		12.2059
		DT		11.2294
		LR		11.1759
		GP		7.9437
		SMOreg		7.4378
		M5P		7.7834
Present Study	https://doi.org/10.5281/zenodo.1209601	RF	RMSE	98%
		LR		62%
		KNN		88%
		K-Fold+ LR		93%
		GWO+ KNN		96.67%

5. CONCLUSION

An increasing number of researchers contribute to the topic of the current study of risk forecasting requirements. This study aims to investigate machine learning methods for risk prediction using a new data set for the first time. This study made the following contributions. For the purposes of risk prediction, three ML algorithms were investigated and then optimized using (Fold-K) and swarm intelligence represented by the gray wolf algorithm was applied. For the proposed model, a comprehensive comparison of different strategies is made. It has been proven that the probability of success or failure of a software project is greatly influenced by the requirements of risk forecasting. A new risk anticipation in the project early on is required to solve this problem. The results showed that the (RF) algorithm excels in prediction as it is more accurate (98%), and the improvement also contributed to raising the accuracy of the LR algorithm by (31%) than its usual accuracy, and (GWO) also contributed to raising the accuracy of the (KNN) algorithm by (5%).

Acknowledgement

The researchers extend their thanks and appreciation to the College of Computer Science and Mathematics, Department of Software Engineering, for providing support in the completion of this work.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

References

1. Z. S. Shaukat, R. Naseem, and M. Zubair, “A dataset for software requirements risk prediction,” *Proc. - 21st IEEE Int. Conf. Comput. Sci. Eng. CSE 2018*, pp. 112–118, 2018, doi: 10.1109/CSE.2018.00022.
2. J. Dhlamini, I. Nhamu, and A. Kachepa, “Intelligent risk management tools for software development,” *Proc. 2009 Annu. Conf. South. African Comput. Lect. Assoc. SACLA 2009*, pp. 33–40, 2009, doi: 10.1145/1562741.1562745.
3. R. Naseem *et al.*, “Empirical assessment of machine learning techniques for software requirements risk prediction,” *Electron.*, vol. 10, no. 2, pp. 1–19, 2021, doi: 10.3390/electronics10020168.
4. D. B. Chattapadhyay, J. Putta, and R. M. R. P, “Risk Identification , Assessments , and Prediction for Mega Construction Projects : A Risk Prediction Paradigm Based on Cross Analytical-Machine Learning Model,” *Build. Environ.*, pp. 1–28, 2021.
5. S. Zhang, Z. Deng, D. Cheng, and X. Zhu, “Author ’ s Accepted Manuscript Efficient k NN Classification Algorithm for Big Data Reference : To appear in : Neurocomputing,” *Neurocomputing*, 2015, doi: 10.1016/j.neucom.2015.08.112.
6. M. H. Mahmud, M. T. H. Nayan, D. M. N. A. Ashir, and M. A. Kabir, “Software Risk Prediction: Systematic Literature Review on Machine Learning Techniques,” *Appl. Sci.*, vol. 12, no. 22, 2022, doi: 10.3390/app122211694.
7. Y. Hu, X. Zhang, X. Sun, M. Liu, and J. Du, “An intelligent model for software project risk prediction,” *2009 Int. Conf. Inf. Manag. Innov. Manag. Ind. Eng. ICIII 2009*, vol. 1, no. X, pp. 629–632, 2009, doi: 10.1109/ICIII.2009.157.
8. A. S. Filippetto, R. Lima, and J. L. V. Barbosa, “A risk prediction model for software project management based on similarity analysis of context histories,” *Inf. Softw. Technol.*, vol. 131, no. March 2020, 2021, doi: 10.1016/j.infsof.2020.106497.
9. A. Iftikhar, M. Alam, R. Ahmed, S. Musa, and M. M. Su’Ud, “Risk Prediction by Using Artificial Neural Network in Global Software Development,” *Comput. Intell. Neurosci.*, vol. 2021, 2021, doi: 10.1155/2021/2922728.
10. M. Assim and M. Hammad, “Software Defects Prediction using Machine Learning Algorithms,” *Int. Conf. Data Anal. Bus. Ind. W. Towar. a Sustain. Econ. 2020*, 2020.
11. G. P and S. Sankaranarayanan, “Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers,” *Comput. Electr. Eng.*, vol. 94, no. October 2020, p. 107362, 2021, doi: 10.1016/j.compeleceng.2021.107362.
12. J. A. Khan, S. Ur, R. Khan, T. A. Khan, I. Ur, and R. Khan, “An Amplified COCOMO-II Based Cost Estimation Model in Global Software Development Context,” *IEEE Access*, vol. 9, pp. 88602–88620, 2021, doi: 10.1109/ACCESS.2021.3089870.
13. A. Iftikhar, S. M. Ali, M. Alam, S. Musa, and M. Mohd, “Analysis of Risk Factors in Global Software Development : A Cross-Continental Study Using Modified Firefly Algorithm,” *Comput. Intell. Neurosci.*, vol. 2022, 2022.
14. A. N. Okon, S. E. Adewole, and E. M. Uguma, “Artificial neural network model for reservoir petrophysical properties : porosity , permeability and water saturation prediction,” *Model. Earth Syst. Environ.*, no. 0123456789, 2020, doi: 10.1007/s40808-020-01012-4.
15. S. M. Janosik, *Data Wrangling with Python*, vol. 42, no. 4. 2016.
16. E. Bisong, *Training a Neural Network*. 2019. doi: 10.1007/978-1-4842-4470-8_29.
17. S. Dankwa and W. Zheng, “Special issue on using machine learning algorithms in the prediction of kyphosis disease: A comparative study,” *Appl. Sci.*, vol. 9, no. 16, 2019, doi: 10.3390/app9163322.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 56 Issue: 05: 2023
DOI10.17605/OSF.IO/2ZJRG

18. V. Muthiah-Nakarajan and M. M. Noel, “Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion,” *Appl. Soft Comput. J.*, vol. 38, pp. 771–787, 2016, doi: 10.1016/j.asoc.2015.10.034.
19. H. Hassan, M. A. Abdel-Fattah, and A. Ghoneim, “Risk Prediction Applied to Global Software Development using Machine Learning Methods,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, pp. 111–120, 2022, doi: 10.14569/IJACSA.2022.0130913.
20. Ž. Vujović, “Classification Model Evaluation Metrics,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.
21. I. C. Dipto, M. A. Rahman, T. Islam, and H. M. M. Rahman, “Prediction of Accident Severity Using Artificial Neural Network: A Comparison of Analytical Capabilities between Python and R,” *J. Data Anal. Inf. Process.*, vol. 08, no. 03, pp. 134–157, 2020, doi: 10.4236/jdaip.2020.83008.
22. H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, 2015, doi: 10.5121/ijdkp.2015.5201.