Journal of Information Technology Research
Volume 15 • Issue 1

Risk Classification in Global Software Development Using a Machine Learning Approach:

A Result Comparison of Support Vector Machine and K-Nearest Neighbor Algorithms

Asim Iftikhar, Institute of Business Management, Karachi, Pakistan & Malaysian Institute of Information Technology, University of Kuala Lumpur, Kuala Lumpur, Malaysia*

Shahrulniza Musa, Universiti Kuala Lumpur, Malaysia

Muhammad Mansoor Alam, Riphah International University, Islamabad, Pakistan & Malaysian Institute of Information Technology, University of Kuala Lumpur, Kuala Lumpur, Malaysia & Multimedia University, Cyberjaya, Malaysia & School of Computer Science, University of Technology Sydney, Australia

Rizwan Ahmed, Institute of Business Management, Pakistan

Mazliham Mohd Su'ud, Universiti Kuala Lumpur, Malaysia

Laiq Muhammad Khan, Institute of Business Management, Pakistan

Syed Mubashir Ali, College of Computing and Information Sciences, Karachi Institute of Economics and Technology

https://orcid.org/0000-0003-2566-7381

ABSTRACT

Software development through teams at different geographical locations is a trend of the modern era, which is not only producing good results without costing a lot of money but also productive in relation to its cost, low risk, and high return. This shift of perception of working in a group rather than alone is getting stronger day by day and has become an important planning tool and part of their business strategy. In this research, classification approaches like SVM and K-NN have been implemented to classify the true positive events of global software development project risk according to time, cost, and resource. Comparative analysis has also been performed between these two algorithms to determine the highest accuracy algorithms. Results proved that support vector machine (SVM) performed very well in case of cost-related risk and resource related risk whereas KNN is found superior to SVM for time-related risk.

KEYWORDS

Global Software Development, Kth Nearest Neighbor, Machine Learning, Risk Management in Global Software Development, Support Vector Machine

INTRODUCTION

Software development environment is shifting from centralized to a dispersed environment so as to offer advantages over the conventional techniques in the recent years (Al-Zaidi & Qureshi, 2017). Success progressively relies upon utilizing software as a competitive weapon. Over 10 years back,

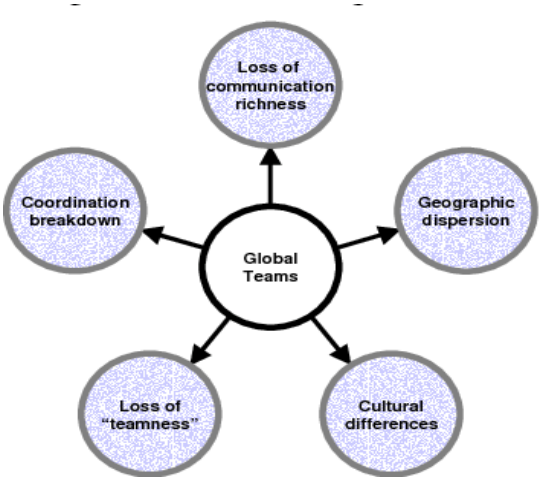
DOI: 10.4018/JITR.299385 *Corresponding Author

looking for lower costs and access to skilled resources, numerous software firms started to explore or experiment with dispersed software development facilities and with outsourcing (Prikladnicki et al., 2003). Therefore, software development is progressively a multisite, multicultural, globally dispersed endeavor. Designers, Engineers, managers, and officials face various, imposing challenges on many levels, from the specialized to the social and cultural (Herbsleb & Moitra, 2001; Prikladnicki et al., 2003). Different scholars name such teams remotely dispersed at various locations or Global Software Development environment (GSD) environment (Iftikhar et al., 2018b).

Global Software Development

The term GSD implies the teams of software experts are scattered, they are located in different geographical locations for the purpose of developing a software on same set of goals and objectives. These teams belong to different cultures and different development backgrounds (Colomo-Palacios et al., 2012). It uses digital communication networks facility to communicate with each other. GSD is popular among the IT organization and a large number of IT employees are taking up global assignment, due to its benefits it offers, irrespective of the duration (Arumugam & Kaliamourthy, 2016). There are some of the significant benefits of GSD that includes continuous development that enhances product quality. It also minimizes costs using cheaper labor and material resources that contributes to increasing productivity. (Al-Zaidi & Qureshi, 2017; AL_Zaidi & Qureshi, 2014; Anjum et al., 2006). In GSD environment distributed teams are still facing many challenges during global software development process such as strategic issues, cultural issues, Inadequate communication, distance, different backgrounds and project and process management issues (Casey & Richardson, 2009; Herbsleb & Moitra, 2001) as shown in Figure 1 (Carmel, 1999).

Figure 1. GSD Challenges (Carmel, 1999)



GSD projects are generally extensive and global evolution steers them to become complicated which in term makes them less probable to succeed. Distributed projects are more likely not to succeed reason being ‘‘physical existence of individuals, time zones, cultural issues, organizational, and stakeholder distances negatively inñuence communication and knowledge exchange between onshore and offshore project team members’’ (Fabriek et al., 2008). When a project is to be executed beyond borders, the assessments of a project manager maximize as the manager now has to take into

consideration the difference in time zones, lingual problems, the overall context and also the built of the specific project (AL_Zaidi & Qureshi, 2014; Hossain et al., n.d.). Global Software development environment made project management task more hectic due to its challenges and complex processes (Colomo-Palacios et al., 2014). Developing software projects to address business needs and requirements in global software development environment is so exceedingly complex and troublesome that it is common for software projects to overrun budgets and exceed scheduled completion dates.

Risk Management

Risk management process is the internal control mechanism driven with certain set of designed practices and procedures in order to properly manage the loopholes within the system, assessment, monitoring and compliance it accordingly (Chadli et al., 2016; Iftikhar et al., 2018a, 2020).

GSD Project based risk are quite uncertain and having less predictability compared to risks involved in collocated software development environment. The basic purpose of diluting the impact of these risks is through cultural, political, different background, communication and coordination and language gap (Galli, 2018).

There are five risk management steps in risk management process (Bhatia & Kapoor, 2011; Nieto-Morote & Ruz-Vila, 2011).

Step 1: Identify the Risk. The task of team is to highlight risks that might affect the project, for which various techniques are used, out of which first is to main a project risk register.

Step 2: Classify the risk. Different risks are grouped together according to their estimated cost or likely impact, probability of occurrence. For example, Credit risk, is classified according to the likelihood of the collection of repayments from the debtor.

Step 3: Analyze the risk. After identification of risk, next important step is to analyze the consequence of each risk, where nature of risk is determined and its capacity to affect project result. This information is also fed into the Project Risk Register.

Step 4: Control the risk. After risk analysis, risk control takes place. It is the method by which software firms evaluate risks and take action to mitigate or eliminate such risks or threats. Which is known as the risk control hierarchy. Eliminating the hazard is the most effective control which must always be aimed at.

Step 5: Review risk control. Ensuring Control measures that have been implemented are effective and efficient. It must be reviewed and revised to make sure they work as planned to determine if any remedial action needs to be taken immediately

Risks Associated with GSD and Risk Factors

Different research scholars addressed the following GSD risk in their researches (Arumugam et al., 2017; Ghaffari et al., 2014; Lopez et al., 2009; Reyes et al., 2011; Verner et al., 2014; Yong et al., 2006)

- Temporal and spatial distance
- Cultural differences (includes attitudes and working styles)
- Use of different software methodologies and different development tools
- Risk factors related to technology
- Project Management Risks
- Project Complexity risks
- Software maintenance issues (includes technical and customer support)

This paper consists of five (5) sections. The first section contains the introduction to this research study. Related work regarding research will be elaborated in Section 2. The machine learning and its techniques used in this paper will be described in Section 3. Section 4 contains the research methodology. Section 5 will discuss the Results and findings and the last section will conclude this study.

RELATED WORK

In the context of risk classification used in software development projects, the authors (Zavvar et al., 2017) proposed an SVM based method and highlighted the importance discussing the factors affecting the risks associated with the classification. Based on the CAR and AUC, the methods of SVM and K-Means were compared with the method proposed in the study. The CAR and AUC in the proposed method are found to be superior as compared to the values of SVM and K-Means. It ultimately contributes to relatively higher precision and better performance of the method proposed for the classification of risk in the software development projects.

In (Mahboob et al., 2017) a solution was found to determine the effort required for a software project based on an organizations historical data for projects. The solution, based on a predictive model, is a result of research which includes two methods i.e., (a) correlation matrix and (b) decision tree. Tests were run using both methodologies, which generated the same results, eventually leading the researchers to identify three parameters that were be used as input for various predictive models. Evaluation from results of these predictive models led to the concluded "Evolutionary Support Vector Machine" as the best model. Therefore, it was determined that the effort required to complete a project can be predicted based on these three parameters (a) number of entities in a project, (b) transaction of the project and (c) project duration in months.

In (Hu et al., 2007) authors determined the most suitable approach for establishing a project's risk evaluation model based on project's complexity. Neural Networks (NN) and Support Vector Machine (SVM) were the two approaches used in this research. Risk experts were interviewed and literature on software risk management was studied to determine six risk categories namely (1) Environment Complexity Risk, (2) Cooperation Risk, (3) Team Risk, (4) Project Management Risk, (5) Project Requirement Complexity Risk and (6) Engineering Risk. Risk factors were used as inputs to predict software project risks and outcomes. Tests were conducted separately using both SVM and standard NN enhanced by GA. Results showed that the latter gave higher accuracy and better risk evaluation model. Hence, it can be safely implied that NN performs better than SVM for projects with complex data relationships.

In (Rong et al., 2016) have proposed a CBA-SVM (where CBA stands for Changing-range Bat Algorithm with Centroid-strategy) software defect prediction model. The CBA-SVM takes advantage of the non-linear computing ability of SVM model and optimization capacity of Bat Algorithm. The simulation results show that this method can get quite a better performance than the other traditional methods and the authors are summarizing their results in where from both, the perspective of accuracy and prediction, the CBA-SVM is best in performance.

Machine Learning

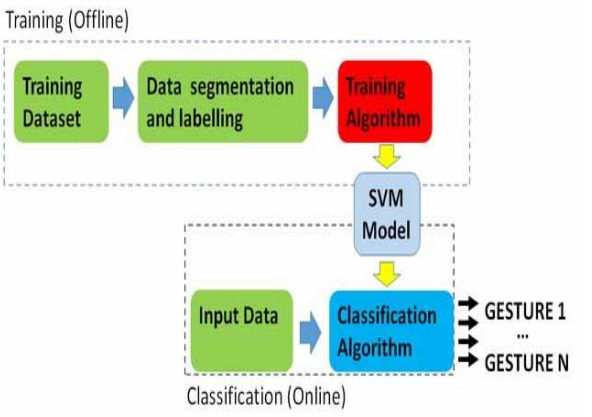
Machine learning is a branch of Artificial Intelligence (AI) that automatically understand and enhance itself with the help of previous experience without any other program. This type of learning specially concentrates on the training of computer programs which can also retrieve the data and make it valuable for themselves. Better decisions in future are made by this learning process which are based on learning with data or observations, like instruction, examples or direct experience. The fundamental objective is to permit the computers that can learn by itself without any assistance and adjust procedures accordingly (Van Liebergen, 2017).

In the past three decades, machine learning (ML), from experimental fascination to realistic technologies over broadly-dispersed commercial applications has progressed dramatically (Jordan & Mitchell, 2015). Machine learning has become the preferred method for designing practical computer vision software systems, voice recognition, natural language processing, robot control and others in artificial intelligence. In several cases, machine learning capacities are applied to a program, like ML and ML applications software systems, methods and libraries that provide ML features (Wan et al., 2019).

Support Vector Machine

Support Vector Machine (SVM) is a special type of supervised machine learning algorithm and one of the classical techniques that can still help solve big data classification problems and regression tasks (Suthaharan, 2016). Classification problems are much more solved by the help of this technique like shown in Figure 2 (Benatti et al., 2017). Each data item that are plot in this algorithm are represented in n-dimensional space (n is the number of features). The value of each element becomes the value of a coordinate, after that classification has been done in that differentiate two classes with the help of finding the hyper-plane. (Iwata et al., 2016).

Figure 2. SVM algorithm block diagram (Benatti et al., 2017)



The classifiers of SVM are Linear, Quadratic, Cubic and Gaussian that uses kernel trick technique to transform the data and then based on these transformations it finds an optimal boundary between the possible outputs.

Linear SVM: In equation 1 Linear SVM classifier is employed where the kernel function of the classifier is given as $K\left(x_{1,}x_{2}\right)$

$$K\left(x_{1,}x_{2}\right)=\left(x_{1}^{T}x_{2}\right) \tag{1}$$

Quadratic SVM: In quadratic SVM lwl has to be minimized. The following quadratic functions 2 to 5 are applied:

$$\min f(n) = \frac{1}{2} \quad w \quad ^{2} \tag{2}$$

$$g(n)=yi*(w.n_{i})–b = 0 \tag{3}$$

$$g(n)=yi*(w.n_{i})–b = 0 \tag{4}$$

$$g(n)=yi*(w.n_i)–b = 0 \quad (5)$$

Cubic SVM: In equation 6 cubic SVM classifier is employed where the kernel function of the classifier is cubic given as $K\left(x_1, x_2\right)$

$$K\left(x_1, x_2\right) = \left(x_1^T x_2\right)^3 \quad (6)$$

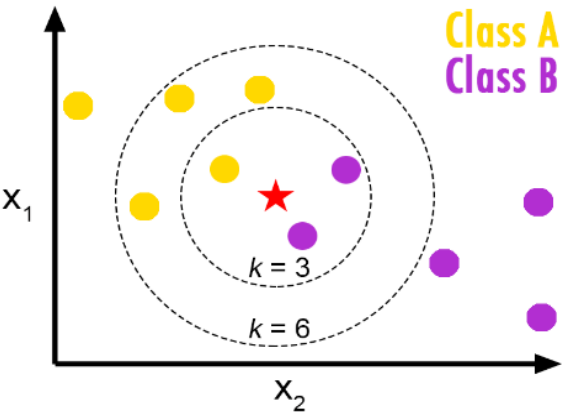
Gaussian SVM: In equation 7 Gaussian SVM classifier is employed where the kernel function of the classifier is Gaussian given as $K\left(x_1, x_2\right)$

$$K\left(x_1, x_2\right) = exp\left(-\frac{x_1 - x_2^2}{2\sigma^2}\right) \quad (7)$$

K-Nearest Neighbor

Another algorithm through which classification and regression problems can be solved is called the K-nearest neighbors (KNN) algorithm. Advantages of this technique is an easy implementation and understanding while the computational cost of this technique is very high due to huge size of data. K-NN was the first technique that used to classify the data at early stage when there was no data provided (Hidmi & Sakar, 2017). KNN classifier is to group unlabeled observations by handing over them to the class of the most comparative labeled examples as shown in figure 3 (Bazmara et al., 2013). For both training and test dataset Characteristics of observations are gathered (Zhang, 2016).

Figure 3. K nearest neighbor classifier procedure (Bazmara et al., 2013)



The classifiers of KNN are Fine, Medium, Coarse, Cosine, Cubic and Weighted which uses data and classify new data points based on similarity measures

Fine and Medium KNN: The Fine and Medium KNN algorithms make use of Euclidian distance function as shown in equation 8 and 9 to determine the nearest neighbors.

$$d = \sqrt{\left(x_1 - y_1\right)^2 + (x_2 - y_2)^2} \tag{8}$$

$$\sqrt{\sum_{i=1}^{k}\left(x_i - y_i\right)^2} \tag{9}$$

Cubic KNN: Cubic KNN classifier uses the cubic distance metric as shown in equation 10

Weighted KNN: Weighted KNN classifier uses the distance weighting as shown in equation 11 to 13

$$\hat{f}\left(x_q\right) \leftarrow \frac{\sum_{i=1}^{k} w_i f\left(x_i\right)}{\sum_{i=1}^{k} w_i} \tag{11}$$

Where

$$\sqrt{\sum_{i=1}^{n} w_i \left|x_i - y_i\right|^2} \tag{13}$$

The two approaches Support Vector Machine and K Nearest Neighbor used in this study have different scopes for application depending upon the data and consequently both the techniques give different results in terms of measures with which the results of two approaches can be compared in terms of accuracy, precision, prediction speed, recall and F-Measure. KNN is more sensitive to the length of the training dataset whereas SVM performs better with small sample scale (Wang et al., 2018)

RESEARCH METHODOLOGY

There are mainly three types of risk factors that are cumulatively linked to an overall risk in a project related to GSD, i.e. time risk, cost risk and resource risk. But the extent of the risk posed by each risk factor may differ in weightage or effect on an overall risk. Therefore, a thorough examination of the three risks is needed to find the extent to which each risk contributes to an overall risk in the project. The method of classification has been used to determine the contribution of each type of risk on the overall risk of the project for the firms involved in the process of software development and located in different geographical regions. To assess this aspect, the firms engaged in GSD located in the United States, Australia and Pakistan are selected. To select the firms, the method of convenience sampling is used due to the time limitation for the completion of this research. To make the findings of this research validated, the representative sample has been selected for data analysis in terms of the firms engaged in similar types of projects pertinent to GSD.

Under umbrella of Support Vector Machine (Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM and Medium Gaussian SVM and Coarse Gaussian SVM) algorithms and under umbrella of K-Nearest Neighbor (Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN and Weighted KNN) algorithms implemented for risk classification in GSD projects to classify the true positive events of risks according to project time, cost and resource. All above mentioned algorithms will calculate Accuracy, Prediction Speed, Training Time, Precision, Recall, F-Measure.

Confusion matrix is used to calculate Accuracy, Precision, Recall and F-measures. For Confusion matrix with True Positive (TP) and False Negative (FN) rates see Appendix

Data Collection

Questionnaire survey was designed to find out the risks concerning the challenges of global software development. The questionnaire developed in this study covers the items related to the three types of risks i.e., Time related, Cost related and Resource related risks that contribute to the overall risks of the GSD projects. The participants had to choose the options from 0 (Very Unlikely), 1 (Unlikely), 2 (Neutral), 3 (Likely) to 4 (Very Likely). The questionnaire was sent to 760 both medium and large size software development organizations. To ensure diversity and bring credibility to survey, questionnaire was not only sent to organizations based in Pakistan but also Australia and the USA. Overall 274 responses were received comprising 103 from Australia, 107 from USA and 64 from Pakistan. Project Mangers, Team leaders, System and Business Analysts participated in the survey. In total there were 390 responses, however 116 responses were rejected since they were incomplete and some organizations had failed to answer certain questions. Data from 274 organizations, as shown in Table 4 and 5 has been trained using Linear Regression and Decision Tree Regression algorithm and obtained the required results. The designed questionnaire was focused to identify risks related to Time, Budget and resources. The data set contains 4 classes named as Low, Moderate, Medium and High that are related to time, cost and resource related risks as shown in Table 1 to 3.

Table 1. Dataset Classes related to Time Related Risks

Risk %Age	Class	Category
36 to 42	0	High
28 to 35	1	Medium
20 to 27	2	Moderate
13 to 20	3	Low

Table 2. Dataset Classes related to Cost Related Risks

Risk %Age	Class	Category
45 to 54	0	High
37 to 44	1	Medium
29 to 36	2	Moderate
21 to 28	3	Low

Table 3. Dataset Classes related to Resource Related Risks

Risk %Age	Class	Category
44 to 53	0	High
35 to 43	1	Medium
26 to 34	2	Moderate
17 to 25	3	Low

Table 4 and 5 reveals the glimpse of the responses received for Q1 to Q30. The results presented in the table identifies that the data contains variability in responses in terms of numbers representing risk categories for all questions except responses to Q4, Q5 and Q7, and it provides rationale to use classification method.

Table 4. Sample Data Set Part-I (from total of 274 Data Set)

Country	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
AUS	1	1	0	1	1	4	2	3	3	3	3	3	4	4	1
AUS	0	0	0	1	1	2	2	3	3	1	1	3	3	1	1
AUS	2	2	1	1	1	1	2	1	3	3	3	1	3	1	1
PAK	2	1	1	1	1	2	2	3	0	3	3	3	3	2	3
PAK	2	2	2	1	1	1	2	2	2	2	1	0	3	3	3
PAK	1	1	1	1	1	2	2	3	1	1	1	3	3	1	1
USA	2	1	0	1	1	1	2	4	3	4	4	3	3	1	0
USA	3	1	0	1	1	1	2	3	2	2	2	3	3	3	3
USA	1	1	0	1	1	2	2	3	3	3	3	4	3	1	3

Table 5. Sample Data Set Part-II (from total of 274 Data Set)

Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	output
3	1	3	4	4	3	3	3	4	4	1	4	4	3	0	3
3	4	3	4	4	3	1	3	4	4	1	4	4	3	0	2
4	3	1	4	4	3	3	4	3	3	1	4	3	1	1	2
4	4	4	3	4	3	3	3	3	4	4	3	3	3	1	3
3	3	3	3	3	3	3	3	3	3	2	3	3	2	0	1
3	3	1	4	4	3	2	3	3	4	2	3	3	2	1	2
4	3	1	4	4	4	4	3	3	3	2	3	3	2	0	3
2	3	4	4	3	3	3	3	3	3	3	3	2	2	0	2
3	3	3	4	4	3	3	3	3	3	1	3	3	1	1	3

RESULTS AND FINDINGS

This study focuses on the comparative analysis of variety of SVM and KNN classifiers used for classification. The classifiers used in SVM such as Linear, Quadratic, Cubic, Fine, Medium and Coarse Gaussian are compared with KNN classifiers namely Fine, Medium, Coarse, Cosine, Cubic and Weighted KNN. The results are compared on the basis of SVM and KNN parameters (Accuracy, Prediction Speed, Training Time, Precision, Recall and F-Measure) that have been calculated in all classifiers of both algorithms as shown in Table 5 to Table 7. Recall, Precision and F-Measure can be calculated using the following equations 14 to 17.

$$Accuracy = \frac{TP + TN}{T\ P + \ F\ N + \ F\ P + \ T\ N} \tag{14}$$

$$Recall = \frac{Tp}{TP + FN} \tag{15}$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

The results of classification of the factor 'Time Related Risk' are presented in Table 6 that indicates the both Medium, Cubic KNN and Weighted KNN classification models give the highest percentage of accuracy (43.8%) with the highest degree of precision (0.35) but with the moderate recall rate of 0.35. The F measure also is the highest in case of these three models.

Table 6. Comparison of SVM and KNN in Time Related Risks

Classification Model	Accuracy	Prediction Speed	Training Time	Recall	Precision	F-Measure
Linear SVM	40.5%	900 obs/sec	20.894 sec	0.31	0.20	0.24
Quadratic SVM	42.7%	840 obs/sec	18.374 sec	0.33	0.30	0.31
Cubic SVM	40.1%	1500 obs/sec	20.136 sec	0.32	0.30	0.30
Fine Gaussian SVM	40.1%	2700 obs/sec	19.842 sec	0.32	0.30	0.30
Medium Gaussian SVM	40.1%	1900 obs/sec	18.858 sec	0.32	0.30	0.30
Coarse Gaussian SVM	40.9%	1600 obs/sec	18.518 sec	0.31	0.21	0.25
Fine KNN	36.5%	1100 obs/sec	57.315 sec	0.41	0.23	0.29
Medium KNN	43.8%	1100 obs/sec	49.24 sec	0.35	0.35	0.35
Coarse KNN	42.7%	6200 obs/sec	57.309 sec	0.33	0.23	0.27
Cosine KNN	42.7%	3900 obs/sec	57.037 sec	0.33	0.23	0.27
Cubic KNN	43.8%	7100 obs/sec	56.456 sec	0.35	0.35	0.35
Weighted KNN	43.8%	8000 obs/sec	56.312 sec	0.35	0.35	0.35

Table 7 gives the results of the fitted classification models on the variable Cost Related Risk. It indicates that all the classification models related to SVM are found superior to KNN based models.

Table 7. Comparison of SVM and KNN in Cost Related Risks

Classification Model	Accuracy	Prediction Speed	Training Time	Recall	Precision	F-Measure
Linear SVM	75.2%	2600 obs/sec	2.6182 sec	0.80	0.78	0.79
Quadratic SVM	75.2%	2400 obs/sec	2.1565 sec	0.80	0.78	0.79
Cubic SVM	75.2%	1900 obs/sec	4.1533 sec	0.80	0.78	0.79
Fine Gaussian SVM	75.2%	2300 obs/sec	4.0201 sec	0.80	0.78	0.79
Medium Gaussian SVM	75.2%	2700 obs/sec	4.9739 sec	0.80	0.78	0.79
Coarse Gaussian SVM	73.4%	2800 obs/sec	4.8453 sec	0.76	0.75	0.75
Fine KNN	65.7%	8700 obs/sec	4.0101 sec	0.66	0.65	0.65
Medium KNN	72.3%	7000 obs/sec	1.6272 sec	0.77	0.73	0.74
Coarse KNN	60.6%	5300 obs/sec	1.2136 sec	0.46	0.31	0.37
Cosine KNN	60.6%	3900 obs/sec	27.156 sec	0.46	0.31	0.37
Cubic KNN	72.3%	7400 obs/sec	3.0574 sec	0.77	0.73	0.74
Weighted KNN	72.3%	6700 obs/sec	2.9334 sec	0.77	0.73	0.74

Table 8. Comparison of SVM and KNN in Resource Related Risks

Classification Model	Accuracy	Prediction Speed	Training Time	Recall	Precision	F-Measure
Linear SVM	50.0%	1900 obs/sec	2.5775 sec	0.51	0.60	0.55
Quadratic SVM	58.0%	2200 obs/sec	2.1376 sec	0.59	0.64	0.61
Cubic SVM	58.0%	1300 obs/sec	4.1596 sec	0.59	0.64	0.61
Fine Gaussian SVM	58.0%	1400 obs/sec	4.0244 sec	0.59	0.64	0.61
Medium Gaussian SVM	58.0%	2900 obs/sec	5.4938 sec	0.59	0.64	0.61
Coarse Gaussian SVM	58.0%	2900 obs/sec	5.3512 sec	0.59	0.64	0.61
Fine KNN	43.8%	9100 obs/sec	10.864 sec	0.47	0.50	0.48
Medium KNN	55.5%	6700 obs/sec	1.8012 sec	0.58	0.59	0.58
Coarse KNN	40.9%	4600 obs/sec	2.4498 sec	0.30	0.35	0.32
Cosine KNN	38.7%	7200 obs/sec	4.5219 sec	0.34	0.29	0.31
Cubic KNN	55.5%	8700 obs/sec	6.0833 sec	0.58	0.59	0.58
Weighted KNN	55.5%	8700 obs/sec	5.7722 sec	0.58	0.59	0.58

The values of all measures such as accuracy, recall, precision and F Measure, on the basis of which models are compared, SVM Models have relatively greater values as compared to KNN classification models.

Table 8 presents the results for the comparison of SVM and KNN methods of classification for the variable Resource Related Risks It reveals that SVM methods are superior as compared to KNN classification methods in terms of accuracy, recall, precision and F measure that are used for

comparison. All the methods employed in SVM except Linear SVM have an accuracy of 58% with recall rate of 0.59, precision is 0.64 whereas F measure has a value of 0.61 for all models.

Considering the difference in results in terms of the classification approach, SVM is found efficient as gives relatively higher values for the variables of Cost Related Risk and Resource Related Risk. Whereas the methods pertinent to KNN gives relatively higher values in case of Cost Related Risk that were found in case of the other two factors related to risks, is found weaker in case of predicting risks related to resource as all measures of KNN have relatively low values as compared to the values of measures of SVM. As the KNN approach is sensitive to outliers it does not give good results if the data have a greater variability and in this case SVM models become superior to KNN classification models. This may be the reason Cost Related Risk and Resource Related Risk that SVM is superior to KNN and vice versa for Time Related Risk.

CONCLUSION

GSD is not a simple software development environment. Organizations face couple of challenges under the umbrella, which should be acknowledged earlier in the implementation process. Since you are managing individuals who are from different cultures, backgrounds, time zones and past project experiences so distributed teams must have good risk management strategy in place. ML based algorithms or techniques give more practical approach than conventional techniques to address risk management in GSD environment. In this research paper classification has been done using SVM and KNN machine learning approaches to classify the true positive events in GSD projects risks according to Time, Cost and Resource. To determine the highest accuracy algorithm, a comparison has also been done. Results proved that SVM gives better results in Cost and Resource related risks and in Time-related risks KNN outperformed SVM.

FUNDING AGENCY

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

Al-Zaidi, A., & Qureshi, R. (2017). Global software development geographical distance communication challenges. *The International Arab Journal of Information Technology*, *14*(2), 215–222.

Al-Zaidi, A. S., & Qureshi, M. R. J. (2014). Scrum practices and global software development. *International Journal of Information Engineering and Electronic Business*, *6*(5), 22.

Anjum, M., Zafar, M. I., & Mehdi, S. A. (2006). Establishing guidelines for management of virtual teams. *IADIS Virtual Multi Conference on Computer Science and Information Systems (Software Engineering and Applications)*.

Arumugam, C., & Kaliamourthy, B. (2016). Global Software development: An approach to design and evaluate the risk factors for global practitioners. *SEKE*, 565–568.

Arumugam, C., Kameswaran, S., & Kaliamourthy, B. (2017). Global software development: A design framework to measure the risk of the global practitioners. *Proceedings of the 7th International Conference on Computer and Communication Technology*, 1–8.

Bazmara, M., Movahed, S. V., & Ramadhani, S. (2013). KNN Algorithm for Consulting Behavioral Disorders in Children. *Journal of Basic and Applied Scientific Research*, *3*, 12.

Benatti, S., Milosevic, B., Farella, E., Gruppioni, E., & Benini, L. (2017). A prosthetic hand body area controller based on efficient pattern recognition control strategies. *Sensors (Basel)*, *17*(4), 869.

Bhatia, N., & Kapoor, N. (2011). Fuzzy cognitive map based approach for software quality risk analysis. *Software Engineering Notes*, *36*(6), 1–9.

Carmel, E. (1999). *Global software teams: Collaborating across borders and time zones*. Prentice Hall PTR.

Casey, V., & Richardson, I. (2009). Implementation of Global Software Development: A structured approach. *Software Process Improvement and Practice*, *14*(5), 247–262.

Chadli, S. Y., Idri, A., Fernández-Alemán, J. L., Ros, J. N., & Toval, A. (2016). Identifying risks of software project management in Global Software Development: An integrative framework. *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, 1–7.

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F. J., & Tovar, E. (2014). Project managers in global software development teams: A study of the effects on productivity and performance. *Software Quality Journal*, *22*(1), 3–19.

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J. (2012). Analyzing human resource management practices within the GSD context. *Journal of Global Information Technology Management*, *15*(3), 30–54.

Fabriek, M., van den Brand, M., Brinkkemper, S., Harmsen, F., & Helms, R. (2008). Reasons for Success and Failure in Offshore Software Development Projects. *ECIS*, 446–457.

Galli, B. J. (2018). Addressing Risks in Global Software Development and Outsourcing: A Reflection of Practice. [IJRCM]. *International Journal of Risk and Contingency Management*, *7*(3), 1–41.

Ghaffari, M., Sheikhahmadi, F., & Safakish, G. (2014). Modeling and risk analysis of virtual project team through project life cycle with fuzzy approach. *Computers & Industrial Engineering*, *72*, 98–105.

Herbsleb, J. D., & Moitra, D. (2001). Global software development. *IEEE Software*, *18*(2), 16–20.

Hidmi, O., & Sakar, B. E. (2017). Software development effort estimation using ensemble machine learning. *Int J Comput Commun Instrum Eng*, *4*(1), 143–147.

Hossain, E., Babar, M. A., & Verner, J. (n.d.). *How Can Agile Practices Minimize Global Software Development Co-ordination Challenges?* Academic Press.

Hu, Y., Huang, J., Chen, J., Liu, M., & Xie, K. (2007). Software project risk management modeling with neural network and support vector machine approaches. *Third International Conference on Natural Computation (ICNC 2007)*, *3*, 358–362.

Iftikhar, A., Musa, S., Alam, M., & Su'ud, M. M. (2020). Artificial Intelligence Based Risk Management in Global Software Development: A Proposed Architecture to Reduce Risk by Using Time, Budget and Resources Constraints. *Journal of Computational and Theoretical Nanoscience*, *17*(2–3), 878–885.

Iftikhar, A., Musa, S., Alam, M., Su'ud, M. M., & Ali, S. M. (2018a). Application of Soft Computing Techniques in Global Software Development: State-of-the-art Review. *International Journal of Engineering & Technology, 7*(4.15), 304–310.

Iftikhar, A., Musa, S., Alam, M., Su'ud, M. M., & Ali, S. M. (2018b). A survey of soft computing applications in global software development. *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 1–4.

Iwata, K., Liebman, E., Stone, P., Nakashima, T., Anan, Y., & Ishii, N. (2016). Bin-Based Estimation of the Amount of Effort for Embedded Software Development Projects with Support Vector Machines. In *Computer and Information Science 2015* (pp. 157–169). Springer.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Lopez, A., Nicolas, J., & Toval, A. (2009). Risks and safeguards for the requirements engineering process in global software development. *2009 Fourth IEEE International Conference on Global Software Engineering*, 394–399.

Mahboob, T., Gull, S., Ehsan, S., & Sikandar, B. (2017). Predictive Approach towards Software Effort Estimation using Evolutionary Support Vector Machine. *International Journal of Advanced Computer Science and Applications*, 446–454.

Nieto-Morote, A., & Ruz-Vila, F. (2011). A fuzzy approach to construction project risk assessment. *International Journal of Project Management*, *29*(2), 220–231.

Prikladnicki, R., Nicolas Audy, J. L., & Evaristo, R. (2003). Global software development in practice lessons learned. *Software Process Improvement and Practice*, *8*(4), 267–281.

Reyes, F., Cerpa, N., Candia-Véjar, A., & Bardeen, M. (2011). The optimization of success probability for software projects using genetic algorithms. *Journal of Systems and Software*, *84*(5), 775–785.

Rong, X., Li, F., & Cui, Z. (2016). A model for software defect prediction using support vector machine based on CBA. *International Journal of Intelligent Systems Technologies and Applications*, *15*(1), 19–34.

Suthaharan, S. (2016). Support vector machine. In *Machine learning models and algorithms for big data classification* (pp. 207–235). Springer.

Van Liebergen, B. (2017). Machine learning: A revolution in risk management and compliance? *Journal of Financial Transformation*, *45*, 60–67.

Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, *56*(1), 54–78.

Wan, Z., Xia, X., Lo, D., & Murphy, G. C. (2019). How does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering*.

Wang, F., Zhen, Z., Wang, B., & Mi, Z. (2018). Comparative study on KNN and SVM based weather classification models for day ahead short term solar PV power forecasting. *Applied Sciences (Basel, Switzerland)*, *8*(1), 28.

Yong, H., Juhua, C., Zhenbang, R., Liu, M., & Kang, X. (2006). A neural networks approach for software risk analysis. *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, 722–725.

Zavvar, M., Yavari, A., Mirhassannia, S. M., Nehi, M. R., Yanpi, A., & Zavvar, M. H. (2017). Classification of risk in software development projects using support vector machine. *Journal of Telecommunication Electronic and Computer Engineering*, *9*(1), 1–5.

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11).

APPENDIX A – ADDITIONAL FIGURES

Figure 4-9. (4) TP and FN Rates of Linear SVM in Time related risks, (5) TP and FN Rates of Quadratic SVM in Time related risks, (6) TP and FN Rates of Cubic SVM in Time related risks, (7) TP and FN Rates of Fine Gaussian SVM in Time related risks, (8) TP and FN Rates of Medium Gaussian SVM in Time related risks, (9) TP and FN Rates of Coarse Gaussian SVM in Time related risks

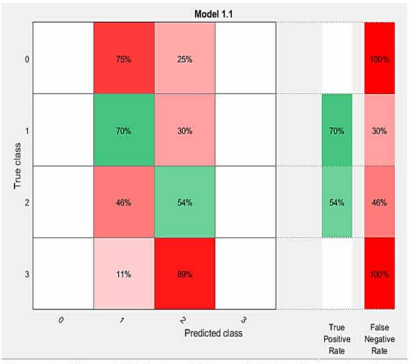


Figure 4. TP and FN Rates of Linear SVM in Time related risks

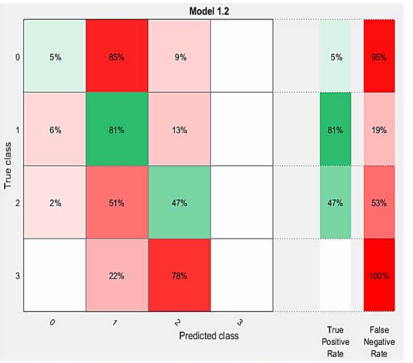


Figure 5. TP and FN Rates of Quadratic SVM in Time related risks

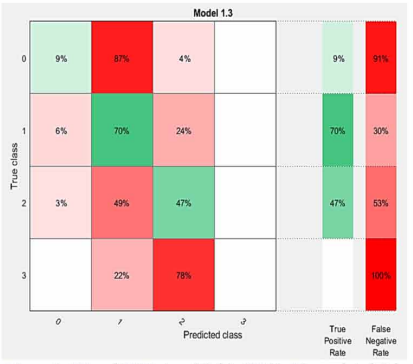


Figure 6. TP and FN Rates of Cubic SVM in Time related risks

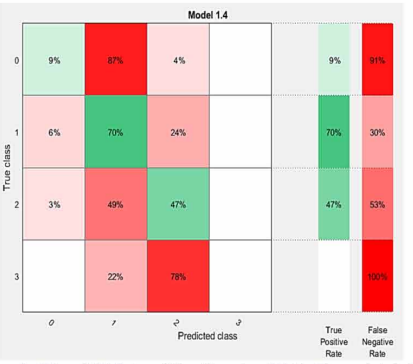


Figure 7. TP and FN Rates of Fine Gaussian SVM in Time related risks

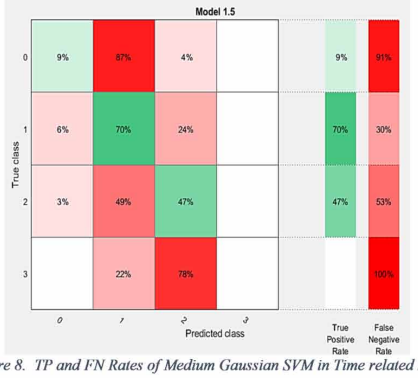


Figure 8. TP and FN Rates of Medium Gaussian SVM in Time related risks

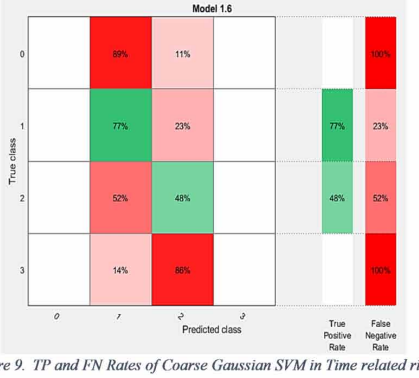


Figure 9. TP and FN Rates of Coarse Gaussian SVM in Time related risks

Figure 10-15. (10) TP and FN Rates of Fine KNN in Time related risks, (11) TP and FN Rates of Medium KNN in Time related risks, (12) TP and FN Rates of Coarse KNN in Time related risks, (13) TP and FN Rates of Cosine KNN in Time related risks, (14) TP and FN Rates of Cubic KNN in Time related risks, (15) TP and FN Rates of Weighted KNN in Time related risks

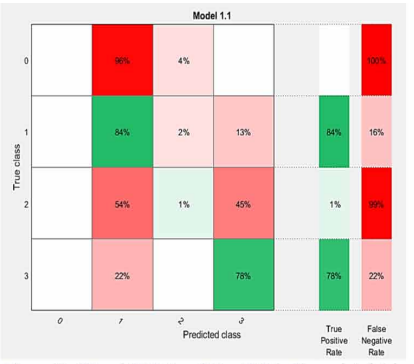


Figure 10. TP and FN Rates of Fine KNN in Time related risks

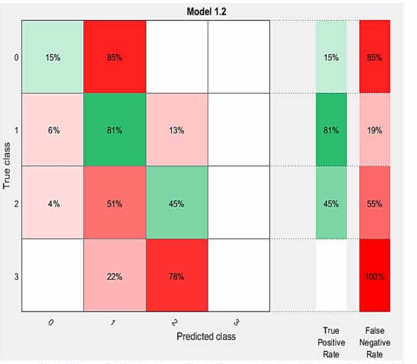


Figure 11. TP and FN Rates of Medium KNN in Time related risks

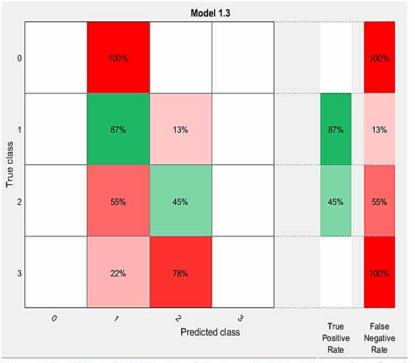


Figure 12. TP and FN Rates of Coarse KNN in Time related risks

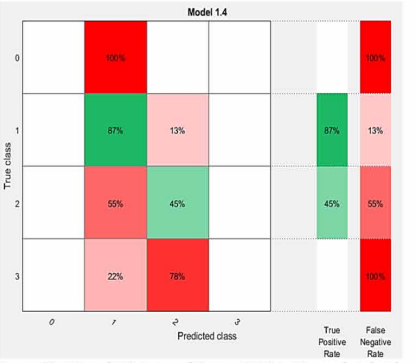


Figure 13. TP and FN Rates of Cosine KNN in Time related risks

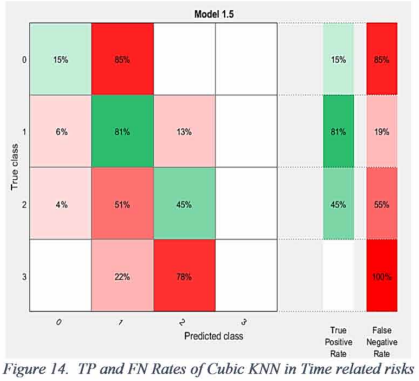


Figure 14. TP and FN Rates of Cubic KNN in Time related risks

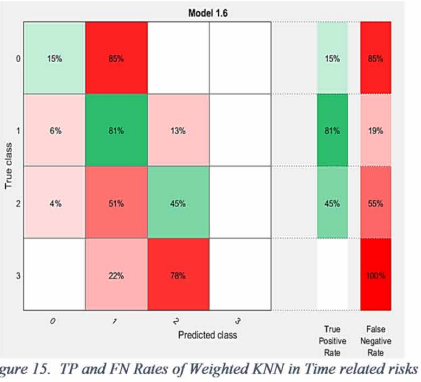


Figure 15. TP and FN Rates of Weighted KNN in Time related risks

Figure 16-21. (16) TP and FN Rates of Linear SVM in Cost related risks, (17) TP and FN Rates of Quadratic SVM in Cost related risks, (18) TP and FN Rates of Cubic SVM in Cost related risks, (19) TP and FN Rates of Fine Gaussian SVM in Cost related risks, (20) TP and FN Rates of Medium Gaussian SVM in Cost related risks, (21) TP and FN Rates of Coarse Gaussian SVM in Cost related risks

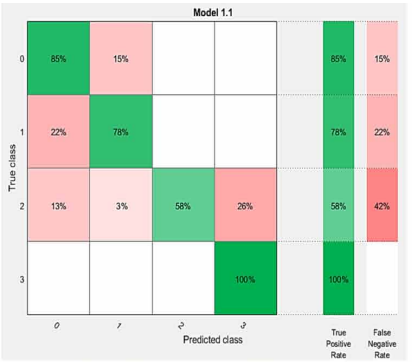


Figure 16. TP and FN Rates of Linear SVM in Cost related risks

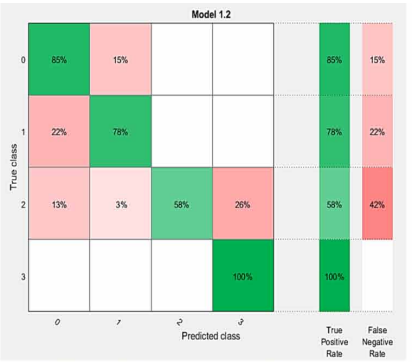


Figure 17. TP and FN Rates of Quadratic SVM in Cost related risks

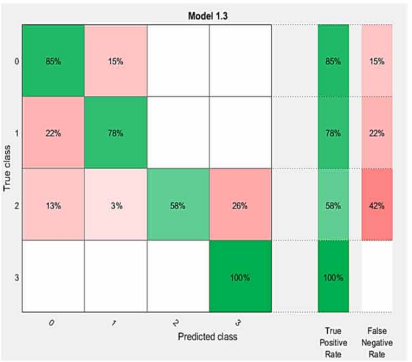


Figure 18. TP and FN Rates of Cubic SVM in Cost related risks

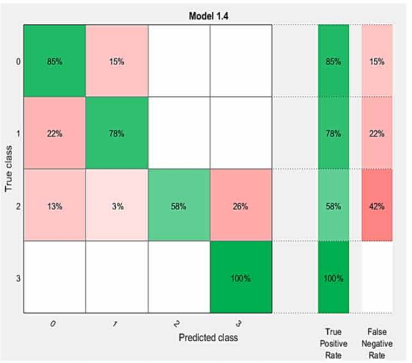


Figure 19. TP and FN Rates of Fine Gaussian SVM in Cost related risks

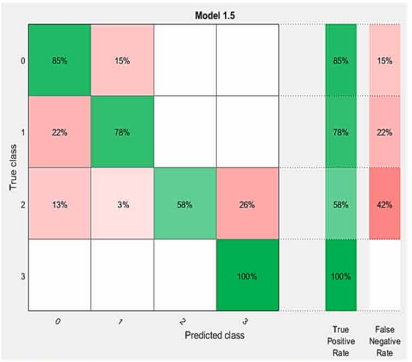


Figure 20. TP and FN Rates of Medium Gaussian SVM in Cost related risks

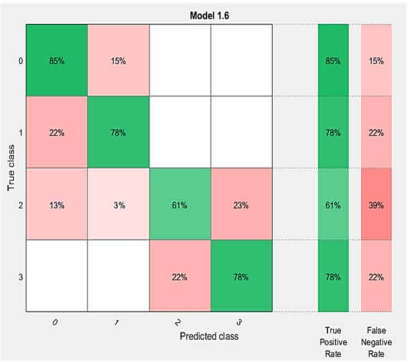


Figure 21. TP and FN Rates of Coarse Gaussian SVM in Cost related risks

Figure 22-27. (22) TP and FN Rates of Fine KNN in Cost related risks, (23) TP and FN Rates of Medium KNN in Cost related risks, (24) TP and FN Rates of Coarse KNN in Cost related risks, (25) TP and FN Rates of Cosine KNN in Cost related risks, (26) TP and FN Rates of Cosine KNN in Cost related risks, (27) TP and FN Rates of Weighted KNN in Cost related risks

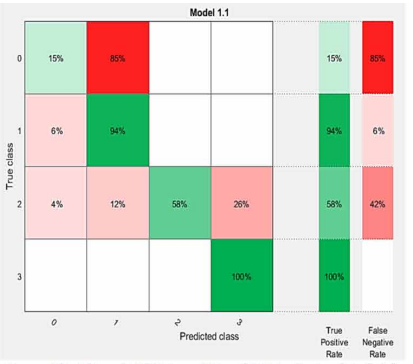


Figure 22. TP and FN Rates of Fine KNN in Cost related risks

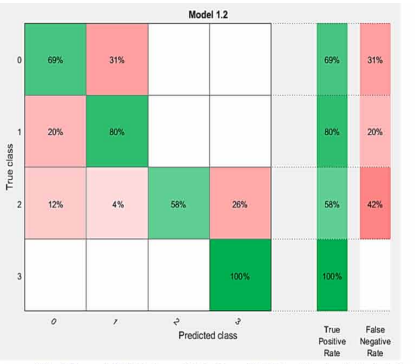


Figure 23. TP and FN Rates of Medium KNN in Cost related risks

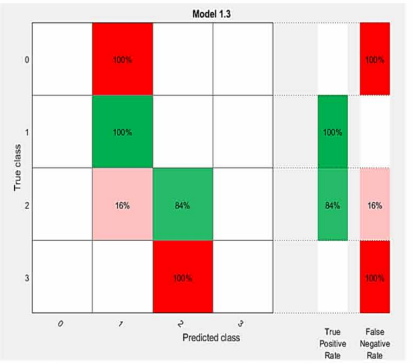


Figure 24. TP and FN Rates of Coarse KNN in Cost related risks

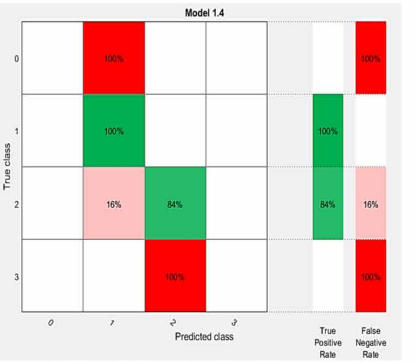


Figure 25. TP and FN Rates of Cosine KNN in Cost related risks

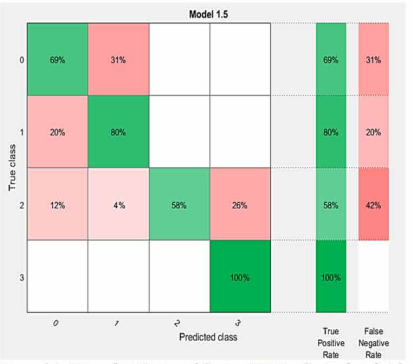


Figure 26. TP and FN Rates of Cosine KNN in Cost related risks

Figure 27. TP and FN Rates of Weighted KNN in Cost related risks

Figure 28-33. (28) TP and FN Rates of Linear SVM in Resource related risks, (29) TP and FN Rates of Quadratic SVM in Resource related risks, (30) TP and FN Rates of Cubic SVM in Resource related risks, (31) TP and FN Rates of Fine Gaussian SVM in Resource related risks, (32) TP and FN Rates of Medium Gaussian SVM in Resource related risks, (33) TP and FN Rates of Coarse Gaussian SVM in Resource related risks

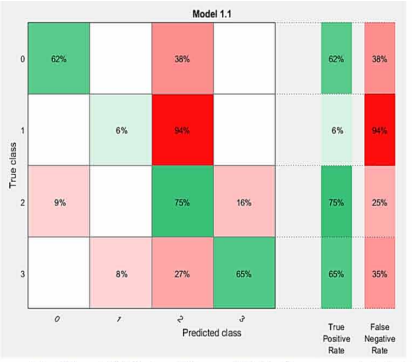


Figure 28. TP and FN Rates of Linear SVM in Resource related risks

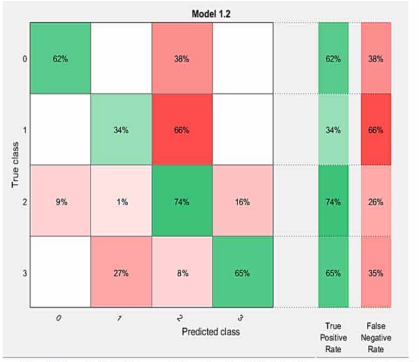


Figure 29. TP and FN Rates of Quadratic SVM in Resource related risks

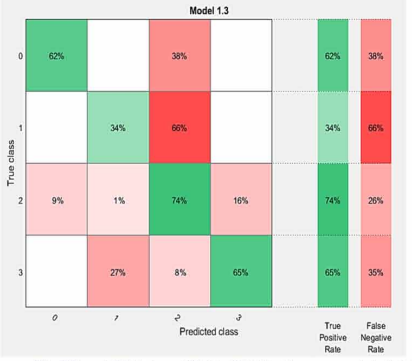


Figure 30. TP and FN Rates of Cubic SVM in Resource related risks

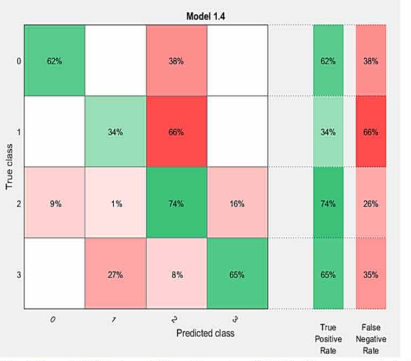


Figure 31. TP and FN Rates of Fine Gaussian SVM in Resource related risks

Figure 32. TP and FN Rates of Medium Gaussian SVM in Resource related risks

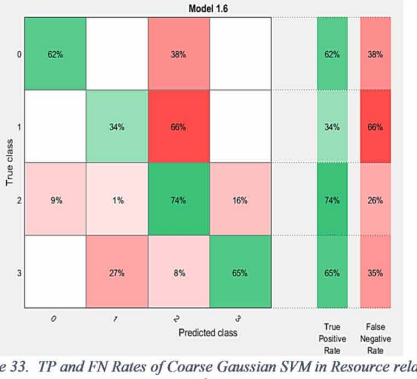


Figure 33. TP and FN Rates of Coarse Gaussian SVM in Resource related risks

Figure 34-39. (34) TP and FN Rates of Fine KNN in Resource related risks, (35) TP and FN Rates of Medium KNN in Resource related risks, (36) TP and FN Rates of Coarse KNN in Resource related risks, (37) TP and FN Rates of Cosine KNN in Resource related risks, (38) TP and FN Rates of Cubic KNN in Resource related risks, (39) TP and FN Rates of Weighted KNN in Resource related risks

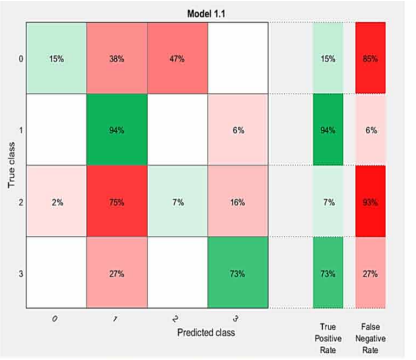


Figure 34. TP and FN Rates of Fine KNN in Resource related risks

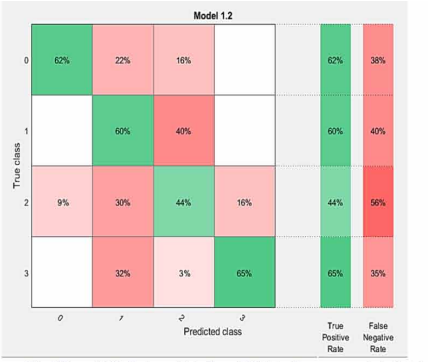


Figure 35. TP and FN Rates of Medium KNN in Resource related risks

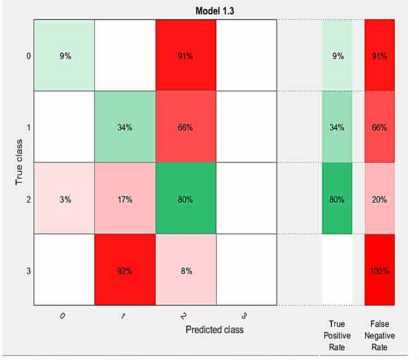


Figure 36. TP and FN Rates of Coarse KNN in Resource related risks

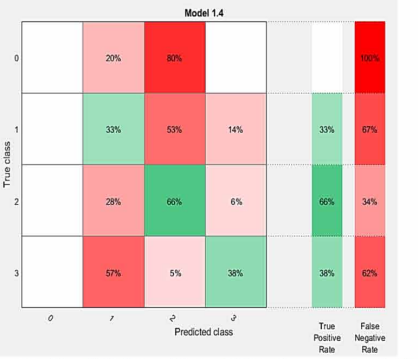


Figure 37. TP and FN Rates of Cosine KNN in Resource related risks

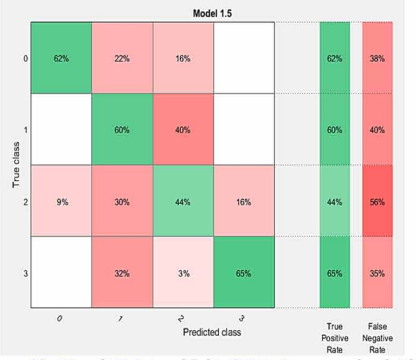


Figure 38. TP and FN Rates of Cubic KNN in Resource related risks

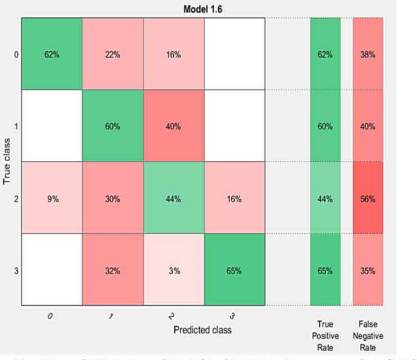


Figure 39. TP and FN Rates of Weighted KNN in Resource related risks

Shahrulniza Musa is a full professor at Malaysian Institute of Information Technology, Universiti Kuala Lumpur (UniKL MIIT). He received his Post-Graduate Diploma in Integrated Research Study in 2005 and Doctor of Philosophy (PhD) in Communication Network Security in 2008, both from the Faculty of Electrical and Electronic Engineering, Loughborough University, UK. His research interest is in Cybersecurity, IoT application, IoT security, BigData Analytic, SDN and Software Engineering. Apart from academic responsibility, he is also active in Software project consultation and development. His ORCID ID and SCOPUS ID are 0000-0003-4867-5085 and_52963995100 respectively.

Muhammad Mansoor Alam received the M.Sc. degree in computer science, the M.S. degree in system engineering, the Ph.D. degree in electrical and electronics engineering, and the Ph.D. degree in computer engineering from France, U.K., and Malaysia, and the Postdoc in machine learning approaches for efficient prediction and decision making from Malaysia. He is a Professor of computer science. He is working as the Associate Dean of CCSIS and the HOD of Mathematics, Statistics and Computer Science Departments. He is also currently working as an Adjunct Professor with UniKL and supervising 12 Ph.D. students. He is enjoying 20 years of research and teaching experience in Canada, England, France, Malaysia, Saudi Arabia, and Bahrain and authored over 150 research articles, which are published in well reputed journals of high impact factor, such as Springer Link book chapters, Scopus indexed journals, and IEEE conferences. He has honor to work as an online laureate (facilitator) for MSIS program run by Colorado State University, USA, and Saudi Electronic University, KSA. He has also established research collaboration with Universiti Kuala Lumpur (UniKL) and Universiti Malaysia Pahang (UMP). Universite de LaRochelle awarded him Très Honorable (with distinction) Ph.D. due to the research impact during the Ph.D.

Syed Mubashir Ali has completed his BS in Computer Engineering from FAST-NUCES, MS in Information Technology from SZABIST, and Ph.D. Computer Science from IoBM in 2006, 2014, and 2020 respectively. Currently he is working as assistant professor at Karachi Institute of Economics and Technology" instead of IoBM. He has several research publications in international peer-reviewed journals and conferences. His current research interest is soft computing, multi-criteria decision making, sustainable supply chain management, circular economy, blockchain, and machine learning.