Multi-Head CNN based Software Development Risk Classification

Original Scientific Paper

Ayesha Ziana M.
Department of Computer Science, Noorul Islam Centre for Higher Education,
Kumaracoil 629180, Tamil Nadu, India
ziananicas@yahoo.com

Charles J.
Department of Software Engineering, Noorul Islam Centre for Higher Education,
Kumaracoil 629180, Tamil Nadu, India
jebapillaicharles@gmail.com

Abstract *– Agile methodology for software development has been in vogue for a few decades, notably among small and medium enterprises. The omission of an explicit risk identification approach turns a blind eye to a range of perilous risks, thus dumping the management into strenuous situations and precipitating dreadful issues at the crucial stages of the project. To overcome this drawback a novel Agile Software Risk Identification using Deep learning (ASRI-DL) approach has been proposed that uses a deep learning technique along with the closed fishbowl strategy, thus assisting the team in finding the risks by molding them to think from diverse perspectives, enhancing wider areas of risk coverage. The proposed technique uses a multi-head Convolutional Neural Network (Multihead-CNN) method for classifying the risk into 11 classes such as over-doing, under-doing, mistakes, concept risks, changes, differences, difficulties, dependency, conflicts, issues, and challenges in terms of producing a higher number of risks concerning score, criticality, and uniqueness of the risk ideas. The descriptive statistics further demonstrate that the participation and risk coverage of the individuals in the proposed methodology exceeded the other two as a result of applying the closed fishbowl strategy and making use of the risk identification aid. The proposed method has been compared with existing techniques such as Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Generalized Linear Models (GLM), and CNN using specific parameters such as accuracy, specificity, and sensitivity. Experimental findings show that the proposed ASRI-DL technique achieves a classification accuracy of 99.16% with a small error rate with 50 training epochs respectively.*

Keywords*: closed fishbowl strategy, explicit risk identification, structured brainstorming, multi-head convolutional neural network*

1. INTRODUCTION

Worldwide circumstances reveal that the success of software development projects highly depends on appropriate planning and an inclination toward minimizing possible risks [1]. Systematic methods for risk management processes and mechanisms are scarce in an agile software development environment [2, 3]. Agile processes perform risk management in an implicit mode by incorporating techniques that handle risks innately. By including time limits and changing requirements, the processes act as risk mitigation techniques, and transparency to the customer improves managing his requirements while also performing risk mitigation [4]. However, the implicit approach is threatening and suitable only as an initial phase and further improvements are needed for sure since a lack of identification of the right risks can lead to poor mitigation, monitoring, and controlling of risks. Explicit risk management strategies that adhere to agile principles are important in agile processes [5,6].

Following explicit risk management strategies in Agile methodology has produced several positive outcomes, like improvements in communication, team efficiency, and product quality [7, 8]. To achieve a certain level of efficiency in Agile Risk Management, an easy methodology that ensures respect for the Agile Manifesto is required [9]. To avoid project failure due to risk traversal, it is necessary to capture and manage risks earlier in the software development process [10]. Recognizing and acknowledging the risks of a project is not always free of complications [11,12].

The ISO/IEC 31000 standard recommends brainstorming as one of the most strongly suitable tools for risk identification and subsequent steps [13]. Brainstorming is suitable for new or substandard procurement tasks since it encourages different and innovative thinking [14]. However, because of its unsystematic nature, there is a chance for potential risks to be overlooked, and the procedure takes more time to obtain the results and arrive at a conclusion [15,16].

Creative teams with diversity in expertise are the key to success in brainstorming [17]. However, they must give up intentional ignorance for team collaboration to be effective [18]. As a result, risk identification assistance tools must consider the above constraints and facilitate communication and collaboration among participants [19-23]. Keeping this in mind, we continue our research to propose a novel Agile Software Risk Identification using the Deep Learning (ASRI-DL) approach [24-27]. The major contributions of the paper are organized as follows.

- The proposed method uses a deep learning technique along with the closed fishbowl strategy, thus assisting the team in finding the risks by molding them to think from diverse perspectives and enhancing wider areas of risk coverage.
- The proposed technique uses a multi-head Convolutional Neural Network (Multihead-CNN) method for classifying the risk into 11 classes in terms of producing a higher number of risks concerning score, criticality, and uniqueness of the risk ideas.
- The effectiveness of the suggested method has been assessed in terms of specific parameters such as accuracy, specificity, and sensitivity.

The remainder of the paper is structured as shown below. The literature review is discussed in Section II. The details of the proposed ASRI-DL scheme are described in Section III. Results and discussion are covered in Section IV. The conclusion and future work are discussed in Section V.

2. LITERATURE REVIEW

In 2018, E. E. Odzaly, [28] highlighted that risk identification was found to be the most difficult process in terms of effort, and people issues are the most crucial ones in Agile software projects; however, automated risk management was not fully achieved through the solution.

In 2018, A. Nadali, et al. [29] proposed a Structured What If Technique (SWIFT), a systematic risk identification study, which clearly illustrates the sources of identified risks, along with their impact, treatment, and countermeasures. The full potential of the model, however, was not investigated.

In 2020, T. E. Abioye, et al. [30] proposed the ontology-based proactive approach towards managing risks by Abioye et al. Though the risk identification process was carried out using multiple techniques this technique was good enough at unearthing almost all risks in the risk identification approach.

In 2020, M. Sousa, [31] proposed a modern board game-based framework that aimed to offer solutions for small enterprises via simple games to stimulate brainstorming sessions. However, the approach requires more time to be spent initially learning the games and then implementing them in the brainstorming sessions, which is a tedious process.

In 2021, M. Nabawy et al. [32], proposed a Risk Breakdown Structure (RBS) a risk identification framework and database, which classifies the risks into threats and opportunities. They present the usage of RBS in systematically identifying and assessing the risks through risk categories.

In 2021, M. Khalilzadeh et al. [33] proposed a fuzzy Delphi method, a combination of the Delphi technique and fuzzy set theory for handling uncertainty, which was used as a risk identification technique to filter the risks previously identified through documentation analysis.

In 2017, M. Zavvar et al. [34] proposed a Support Vector Machine (SVM) to model risk classification in software development projects. According to the experimental results, the proposed method exhibits superior CAR and AUC.

In 2022, M. Yang et al. [35] proposed a multi-layer perceptron model that can predict risks in software development. The experimental findings show that the prediction accuracy is 83.11%, which is higher than that of typical machine learning models.

In 2015 B. Esmaeili et al. [36] investigated the validity of applying these fundamental risk factors to predict safety outcomes. The modeling technique consists of two steps: (1) doing principle components analysis to minimize dimensionality and (2) using principal components as generalized linear models to describe the probability of various risk categories.

In 2022, Q. Wang et al. [37] presented a convolutional neural network (CNN)-based automated technique for identifying software vulnerabilities. CNN is used to classify vulnerabilities automatically. The suggested approach outperformed the competition in macro recall, macro precision, and macro F1 score in experimental comparison and analysis.

Roughly more than 500 risks from various studies were utilized for the objective, some of which are included here in Table 1, we can identify the implicit and explicit risk identification strategies applied in Agile software development projects. As far as we have reviewed from the literature, the fishbowl strategy has not been utilized in software engineering practices, in the risk identification process of brainstorming in Agile software enterprises. This influenced our ideology in applying the ASRI-DL technique to acquire the desired outcome.

Table 1. Practices for Risk Identification in Agile Projects

Practices	References
Daily Standup Meetings	[10, 28]
Sprint Planning	[10, 3, 23, 11]
Brainstorming	[23, 22, 9]
Checklist	[22]
Weekly Sprint Meetings	[16]
Increment	[17]
Prototype	[17]
Product backlog refinement meeting	[17, 28, 4]
Weekly risk meeting	[17]
Technical specification	[17]
Risk Identification Meetings	[23]
Sprint backlog	[28]
Sprint review	[14, 3, 4, 18]
Sprint Retrospective	[3]

3. AGILE SOFTWARE RISK IDENTIFICATION USING DEEP LEARNING (ASRI-DL) FRAMEWORK

The proposed Agile Software Risk Identification using Deep Learning (ASRI-DL) approach incorporates systematic brainstorming for risk identification.

To soar the partaking of the participants in the discussion, the Closed Fishbowl Strategy was lodged, then, the deep learning technique, multi-head CNN takes the input such as human, organizational, technical, non-technical, and capabilities from the fishbowl strategy and classifies it into 11 categories for efficient risk classification.

The block diagram for the proposed ASRI-DL technique is depicted in Fig 1.

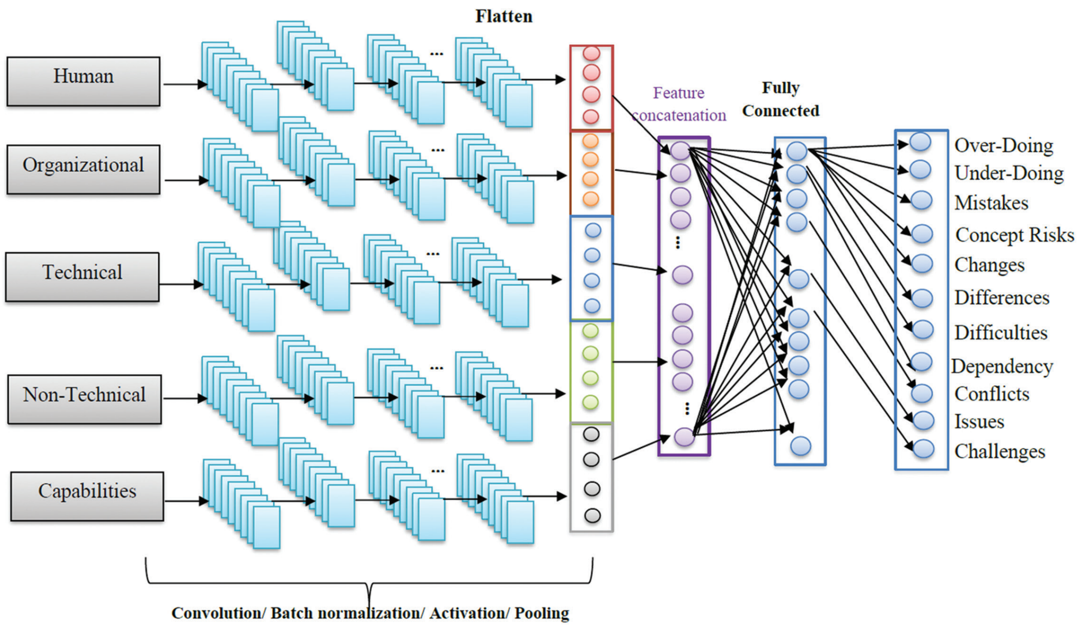

Fig. 1. Architecture of the Proposed ASRI-DL Multi-faceted Taxonomy Assistance for Risk Identification

3.1. MULTI-HEAD CONVOLUTIONAL NEURAL NETWORK

Multi-head convolution (MHCNN) is a CNN in which each time series is handled by a different convolution known as a convolutional head. The proposed method takes data such as humans, organizational, technical, non-technical, and capabilities as input. The three categories *Human*, *Organizational*, and *Technical*, were affixed to the taxonomy for identifying the risk. The Non-technical category is ascended with the Environmental, Marketing, and Legal subcategories. The proposed architecture uses a sliding window to process the test case. The network recovers features based on each phase using window-based time series processing. Convolution is defined as follows:

$$O = \sum_{x=1}^{L} \sum_{y=1}^{W} z(x,y) f(x,y) \quad (1)$$

Where O is the output from $z(x,y)$ and the filter $f(x,y)$ with the length (L) and width (W) respectively. Each temporal sequence receives its own feature map. Equation (2) calculates the number of characteristics in each layer of a typical MHCNN.

$$G = f_m * f_z * d + bias \quad (2)$$

In this case, f_m denotes the number of filters, f_z the filter size, and d the final dimension of the resultant vector from the preceding layer. Each convolution head in MH-CNN requires a 4-dimensional input, which is calculated as follows:

$$I = (n_s, W_s, W_l, n_c) \quad (3)$$

where n_s denote the number of samples in the batch, and n_c refers to the number of channels. The total number of windows is calculated as follows:

$$W_n = \frac{D_p - W_l}{W_t} + 1 \quad (4)$$

The total number of data points in the time series is denoted by D_p, the window length by W_l, and the window step by W_t. As a result, the suggested technique may collect useful information from the local to the global levels. A loss function is cross entropy and the cross-entropy loss function is calculated as follows:

$$loss = -\frac{1}{S}\sum_{n=1}^{N}[y_t \log \hat{y}_t + (1 - y_t)\log(1 - \hat{y}_t)] \quad (5)$$

where S denotes the number of samples and y_t is the probability of a true label, $\hat{y}_t$ is the probability of the predicted label. The proposed method will classify the risk into 11 classes such as over-doing, under-doing, mistakes, concept risks, changes, differences, difficulties, dependency, conflicts, issues, and challenges.

The proposed *Risk Nature* facet is much more stipulated towards investigating the type of risk focusing on the causation. We used a few more attributes to the facet termed *Changes*, *Differences*, *Challenges*, and *Dependency*. Over-Doing (OD) risks arise as a consequence of adding blameworthy elements to the process. The exclusion of obligatory features gives birth to Under-Doing (UD) risks. Erroneous execution of the right concept is the main reason for Mistakes (M) to arise. Conceptual Problems (C) indicate risks originating from an invalid treatment.

We realized the significance of changes since they can occur at any point in time throughout the project cycle and play a huge part in the winning or losing moments of a software project. The significance of *differences*, otherwise known as non-congruence, and software failure is listed as one of its consequences. The measure of *congruence* is computed by the difference between expectations and requirements. In other words, *dependency* may be a cause for dissimilarities or differences. This precipitated us to move the two attributes, difference and dependency, to the Risk Nature facet under the assumption that they might induce risks. *Challenges* allude to coming across new circumstances, ideas that induce cognitive discrepancies, or the catalyst for thriving and development.

Concerning the suitability of the complexity science theory in project management, *complexity* also known as *difficulty*, is a major factor in the failure of big projects. Furthermore, sudden unexpected events, complexities, and clashes were encountered as risks in the literature, which impelled us to categorize them appropriately under the nature of *issues*, *difficulties*, and *conflicts*. To validate the technique, we implemented the Utility Demonstration approach through quasi-experimental research and classifying the risks, which in turn was used as the training material in the experimental session.

4. RESULTS AND DISCUSSION

4.1. SAMPLING

Two small and medium enterprises, C1 and C2, which follow a hybrid methodology of software development, were chosen for the experiment, and the choice was based on the availability and willingness of the participants. Three groups were chosen with unequal sample sizes of 8, 10, and 12, respectively, excluding the facilitators, and a minimum sample size of 30 altogether was ensured. Each group had unique participants. It should be noted that Groups 2 and 3 are from the same company. The number of participants was purely based on the availability of samples from both companies. We ensured that we got at least 30 samples altogether before the experiments, excluding the facilitators, to produce statistically significant results.

Regarding the disparity in each group,

i. Our group 1 initially contained 9 available samples, including the facilitator during the training sessions. However due to a sudden dropout on the day of the experiment, one of the authors was made the facilitator, and the size was reduced to 8.

ii. (Our Group 2 and Group 3 are from the same company, and we were given 24 samples, including 2 facilitators. Among those 22 participant samples available, since our proposed closed fishbowl strategy requires at least four participants in the inner circle along with the facilitator and the remaining must be in the outer circle, we thought 12 participants and 1 facilitator would be ideal for conducting the session in three rounds for our Group 3. Thus, the group was allotted the remaining participants.

4.2. EXPERIMENTAL PROCEDURE

Initially, all three groups were instilled with the basics of Agile methodology and the significance of risk identification in Agile enterprises through oral and visual presentations. Before the final experiment, Groups 1 and 3 acquired a training session with the fishbowl strategy and the proposed ASRI-DL methodology, respectively, which included training with the solved material, hands-on training, and trial experiments. Group 2 was exempt from training as it was our control group. However, a trial experiment was carried out ahead of time.

4.3. EXPERIMENTAL SESSIONS ILLUSTRATION

Three different project scenarios were given as inputs to the three groups to capture the risks. Sessions 1 and 3 were held uninterrupted for around 40 minutes, as scheduled. Session 2 lasted approximately 33 minutes, while the actual planned session was approximately 40 minutes. The reason for this is that the participants felt that enough ideas had been generated and that their ability to discover new risks had slowed to the point where we had no choice but to end the session.

The facilitators were given the goal of directing their team towards yielding as many risks as possible for the given project scenarios, which in turn were evaluated in terms of novelty and their prominence in the project, otherwise termed Score and Criticality. The outcomes of the three experiments were evaluated by eight experts with significant working experience, from the company C2 using a 5-point Likert scale that took the ordinal values of "Very low, Low, Medium, High, and Very high" to assess both the Score and Criticality. The demographic details of the eight experts are given in Table 2.

Table 2. Demographic details of the eight experts

EMPID	Designation	Experience	Evaluated group
EMP1	Technical Architect	2.3 yrs	1
EMP2	Senior Developer	4.8 yrs	1
EMP3	Senior business analyst	3.7 yrs	1,2
EMP4	Module lead	5.2 yrs	2
EMP5	Senior Developer	6.8 yrs	2
EMP6	Senior business analyst	2.8 yrs	3
EMP7	Senior Developer	4.9 yrs	3
EMP8	Senior Developer	4.4 yrs	3

4.4. STATISTICAL ANALYSIS

At the juncture of the actual experiments, Group1 spotted 126 risks; Group2 figured out 83 ideas, and finally, the experiments were put to an end by Group3 by ascertaining 226 risks. For quantitative analysis, the IBM SPSS Statistics Version 26 was used. The values of the ordinal variables 'category', 'subcategory', and 'risk nature' were computed from the median and mode of the ranks imparted by three of our evaluation experts. According to the results, the *Organizational* category tops the list, followed by Human and Capability categories. To eliminate the bias, the two pivotal dependent variables, '*actualscore*' and '*actualcriticality*' for each identified risk is employed by using the formula below:

$$actualscore = uniqueness / totalscores \tag{6}$$

$$actualcriticality = (category * subcategory * risknature) * uniqueness / totalcriticality \tag{7}$$

where category, subcategory, and risk nature denote the recoded rank values proffered by the experts. The total scores and total criticality for each identified risk are computed by summing up the marks allotted by the experts.

4.5. PROPOSITIONS

The narration of the propositions formulated for the study is as follows.

4.5.1. Proposition 1

Null Hypothesis: There is no difference in the distribution of unique ideas with good scores among the three groups.

Alternate Hypothesis: There is a difference in the distribution of unique ideas with high scores among the three groups.

We tested the normality of the distribution by the Shapiro-Wilk test and derived the significance value as 0.000 less than 0.05, and the hypothesis of normality was rejected. The Kruskal Wallis test was conducted to examine the hypothesis, and the result shows that the null hypothesis is rejected since the significance value is 0.000 less than 0.05, which is shown in Table 3. There was also no significant difference between Groups 1 and 2, as assured by the significance value of 0.345 greater than 0.05.

Table 3. Pairwise Comparisons of Experimental Groups for Actual Scores of Identified Risks

Sample1-Sample2	Test-Statistic	Std.Error	Std.Test Statistic	Sig.	Adj.Sig.
Group1-Group2	-15.653	16.579	-.944	.345	1.000
Group1-Group3	-61.565	13.039	-4.722	.000	.000
Group2-Group3	-45.912	15.052	-3.050	.002	.007

4.5.2. Proposition 2

Null Hypothesis: There is no difference in the distribution of unique ideas with good criticality among the three groups.

Alternate Hypothesis: There is a difference in the distribution of unique ideas with good criticality among the three groups.

Table 4. Pairwise Comparisons of Experimental Groups for Actual Criticality of Identified Risks

Sample1-Sample2	Test-Statistic	Std. Error	Std. Test Statistic	Sig.	Adj. Sig.
Group2-Group1	1.798	16.616	.108	.914	1.000
Group2-Group3	-57.819	15.086	-3.833	.000	.000
Group1-Group3	-56.021	13.068	-4.287	.000	.000

Scrutinizing the Kruskal Wallis test results interprets in favor of the alternative hypothesis by rejecting the null hypothesis as the significance value is 0.000 less than 0.05, which is shown in Table 4.

4.6. OTHER FINDINGS

The frequency with which participants engendered their viewpoints in their respective sessions is depicted by the graphical illustration in Fig. 2. It is pretty obvious that the graph tends to be more or less uniform among the participants of Group 3, in contrast to the rest of the groups.

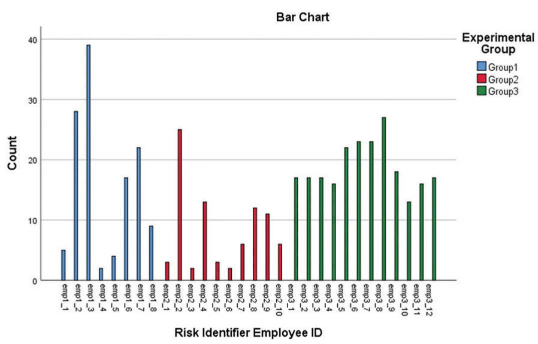

Fig. 2. Bar Chart representing the frequency of participation among the samples across the three groups

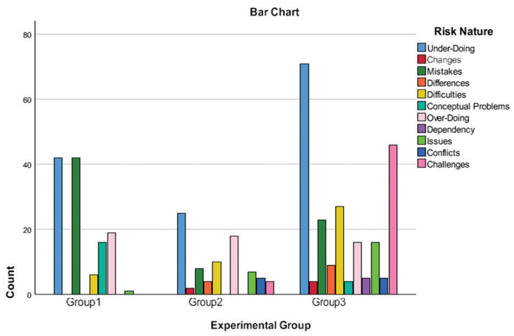

Fig. 3. Bar Chart representing the 'Risk Nature' of the identified risks among the three groups

The cross-tabulation exemplified via the bar chart in Fig. 3 exhibits the rendering of risks under the 'Risk Nature' facet by the 3 groups during their respective brainstorming sessions. Group 3 was the only one to accomplish ideas under every nature of risk. This is a positive sign concerning our proposed methodology.

4.7. FEEDBACK ASSESSMENT

For feedback assessment, a short questionnaire was fabricated and supplied to the participants of the treatment groups at the end of sessions 1 and 3, including the facilitator and one of the dropouts from session 1.

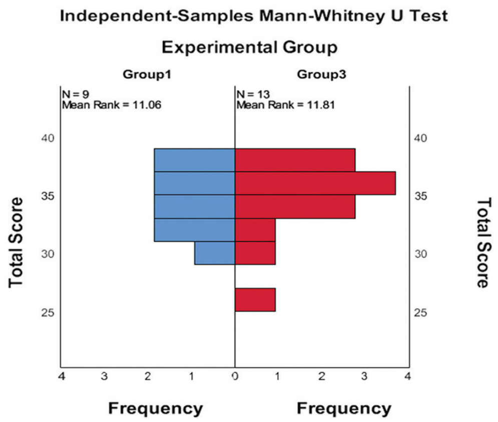

Fig. 4. Mann-Whitney U Test for comparing the mean ranks of the Total Scores in Feedback among Groups 1 and 3

The 'Total Score' is computed by summing up the variables aforementioned. The Mann-Whitney U Test in Fig. 4 reveals the inference of a similar distribution of Total Scores among the 2 groups through the significance value of 0.794 greater than 0.05, thus accepting the null hypothesis.

4.8. RELIABILITY AND VALIDITY ANALYSIS

The reliability analysis for the experiment was attained by the statistical test using Cronbach's Alpha test, as manifested in Tables 5 and 6. As a result of the Cronbach's Alpha values of 0.753 greater than 0.6 and 0.767 greater than 0.6, our experiment is acknowledged to be reliable.

Table 5. Reliability Statistics for Criticality of Risks

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.753	.761	3

Table 6. Reliability Statistics for Score of Risks

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.767	.803	3

The correlation analysis was performed using Spearman's correlation coefficient which is given in Table 7 which indicates a positive correlation between *risk nature* and *actual criticality*.

Table 7. Construct Validity Analysis using Spearman's Rho correlations

Risk types	Risk Nature	Actual Criticality
Correlation Coefficient	1.000	.180
Sig.(2-tailed)	.	.000
N	435	435
Correlation Coefficient	.180	1.000
Sig.(2-tailed)	.000	.
N	435	435

4.9. PERFORMANCE ANALYSIS

In Fig. 5, the accuracy graph is estimated with the 50 epochs and accuracy range. The proposed Multihead-CNN obtains a high accuracy of 99.16% with a small error rate with 50 training epochs in the identification of risks.

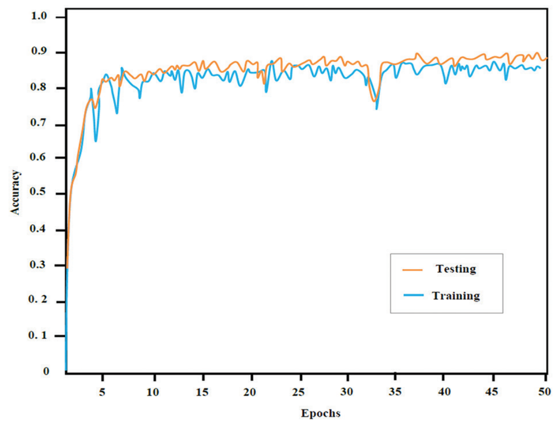

Fig. 5. Training and testing accuracy of the proposed Multihead-CNN model

The proposed multi-head CNN network has been compared with existing techniques such as SVM [34], MLP [35], GLM [36], and CNN [37] using specific parameters such as accuracy, specificity, and sensitivity which are shown in Fig 6. From the figure, it is clear that the proposed method achieves better accuracy than existing techniques.

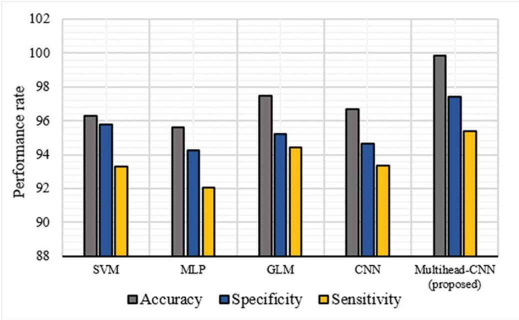

Fig. 6. Performance comparison of the proposed method with existing techniques

The proposed multi-head CNN network has been compared with existing techniques such as SVM [34], MLP [35], GLM [36], and CNN [37] using specific parameters such as accuracy, specificity, and sensitivity which are shown in Fig. 6. From the figure, it is clear that the proposed method achieves better accuracy than existing techniques.

5. CONCLUSION

In this paper, a novel well-defined proactive ASRI-DL methodology method was proposed for risk management, which employs a closed fishbowl strategy and multihead-CNN technique for the risk identification process of Agile software projects. The proposed method has been compared with existing techniques such as SVM, MLP, GLM, and CNN using specific parameters such as accuracy, specificity, and sensitivity. Experimental findings show that the proposed ASRI-DL technique achieves a classification accuracy of 99.16% with a small error rate with 50 training epochs respectively. On this note, we conclude the paper with the future intention of steering the research towards further phases of risk assessment, including risk analysis and prioritization.

6. REFERENCES

[1] M. Esteki, T. J. Gandomani, H. K. Farsani, "A risk management framework for distributed scrum using PRINCE2 methodology", Bulletin of Electrical Engineering and Informatics, Vol. 9, No. 3, 2020, pp. 1299-1310.

[2] B. Prakash, V. Viswanathan, "Distributed cat modeling based agile framework for software development", Sadhana, Vol. 44, No. 7, 2019, pp. 1-11.

[3] C. R. Nelson, G. Taran, L. D. L. Hinojosa, "Explicit Risk Management in Agile Processes", Proceedings of the International Conference on Agile Processes and Extreme Programming in Software Engineering, 2008, pp.190-201.

[4] M. Vieira, J. C. R. Hauck, S. Matalonga, "How Explicit Risk Management is Being Integrated into Agile Methods: Results from a Systematic Literature Mapping", Proceedings of the 19th Brazilian Symposium on Software Quality, Association for Computing Machinery, Vol. 15, 2020, pp. 1-10.

[5] G. P. Gasca-Hurtado, M. C. Gómez-Alvarez, M. Muñoz, A. Peña, "A gamified proposal for software risk analysis in agile methodologies", Proceedings of the European Conference on Software Process Improvement, 2019, pp. 272-285.

[6] S. Patil, R. Ade, "A software project risk analysis tool using software development goal modeling approach", Information Systems Design and Intelligent Applications, 2015, pp. 767-777.

[7] J. T. Marchewka, "A framework for identifying and understanding risks in information technology projects", Journal of International Technology and Information Management, Vol. 19, No. 1, 2010, p. 2.

[8] S. Chaouch, A. Mejri, S. A. Ghannouchi, "A framework for risk management in Scrum development process", Procedia Computer Science, Vol. 164, 2019, pp. 187-192.

[9] B. G. Tavares, M. Keil, C. E. S. da Silva, A. D. de Souza, "A risk management tool for agile software development", Journal of Computer Information Systems, Vol. 61, No. 6, 2021, pp. 561-570.

[10] B. G. Tavares, C. E. S. da Silva, A. D. de Souza, "Practices to improve risk management in agile projects", International Journal of Software Engineering and Knowledge Engineering, Vol. 29, No. 03, 2019, pp. 381-399.

[11] N. Anand, S. Pujar, S. Rao, "A heutagogical interactive tutorial involving Fishbowl with Fish Battle and Round Robin Brainstorming: A novel syndicate metacognitive learning strategy", Medical Journal Armed Forces India, Vol. 77, 2021, pp. S73-S78.

[12] S. de Souza Lopes, R. C. G. de Souza, A. D. G. Contessoto, A. L. de Oliveira, R. T. V. Braga, "A Risk Management Framework for Scrum Projects", Proceedings of the 23rd International Conference on Enterprise Information Systems, Vol. 2, pp. 30-40.

[13] S. Grishunin, S. Suloeva, T. Nekrasova, E. Burova, "Development of Risk Controlling Mechanism and

Tools for Agile Projects in Telecommunications", Proceedings of Internet of Things, Smart Spaces, and Next Generation Networks and Systems, 2020, pp. 270-284.

[14] I. Ermiwin, S. R. Mertosono, Z. Hasyim, "Applying fishbowl technique to improve students speaking skill", E-Journal of ELTS (English Language Teaching Society), Vol. 7, No. 3, 2019.

[15] S. Filyppova, I. Bashynska, B. Kholod, L. Prodanova, L. Ivanchenkova, Ivanchenkov, V, "Risk Management Through Systematization: Risk Management Culture", International Journal of Recent Technology and Engineering, Vol. 8, No. 3, 2019, pp. 6047-6052.

[16] P. Marnada, T. Raharjo, B. Hardian, A. Prasetyo, "Agile project management challenge in handling scope and change: A systematic literature review", Procedia Computer Science, Vol. 197, 2022, pp. 290-300.

[17] M. Nabawy, G. Ofori, M. Morcos, C. Egbu, "Risk identification frame-work in construction of Egyptian mega housing projects", Ain Shams Engineering Journal, Vol. 12, No. 2, 2021, pp. 2047-2056.

[18] O. Ņikiforova, K. Babris, J. Kristapsons, "Survey on Risk Classification in Agile Software Development Projects in Latvia", Applied Computer Systems, Vol. 25, No. 2, 2020, pp. 105-116.

[19] M. Shameem, R. R. Kumar, M. Nadeem, A. A. Khan, "Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process", Applied Soft Computing, Vol. 90, No. 6, 2020, p. 106122.

[20] S. V. Shrivastava, U. Rathod, "A Goal-driven Risk Management Approach for Distributed Agile Development Projects", Australasian Journal of Information Systems, Vol. 23, 2019.

[21] J. Sithambaram, M. H. N. B. M. Nasir, R. Ahmad, "Issues and challenges impacting the successful management of agile-hybrid projects: A grounded theory approach", International Journal of Project Management, Vol. 39, No. 5, 2021, pp. 474-495.

[22] C. Tam, E. J. da Costa Moura, T. Oliveira, J. Varajao, "The factors influencing the success of on-going agile software development projects", International Journal of Project Management, Vol. 38, No. 3, 2020, pp. 165-176.

[23] A. A. Mohamed, N. R. Darwish, H. A. Hefny, "Multiple Linear Regression for Determining Critical Failure Factors of Agile Software Projects", International Journal of Intelligent Engineering and Systems, Vol. 12, No. 3, 2019.

[24] E. D. Canedo, A. P. M. do Vale, R. M. Gravina, R. L. Patrão, L. C. de Souza, V.E. dos Reis, F. L. L. de Mendonça, R. T. de Sousa Jr, "An Applied Risk Identification Approach in the ICT Governance and Management Macroprocesses of a Brazilian Federal Government Agency", Proceedings of the International Conference on Enterprise Information Systems, 2022, pp. 272-279.

[25] D. Kasap, M. Kaymak, "Risk Identification Step of the Project Risk Management", Proceedings of the Portland International Conference on Management of Engineering & Technology, Portland, OR, USA, 5-9 August 2007, pp. 2116-2120.

[26] I.A. Kiral, Z. Kural, S. Comu, "Risk Identification in Construction Projects: Using the Delphi Method", Proceedings of the 11th International Congress on Advances in Civil Engineering, Istanbul, Turkey, 2014.

[27] A. Kobo-Greenhut, H. Reuveni, I. B. Shlomo, R. Megnezi, "Unstructured brainstorming is not enough: structured brainstorming based on four verification and validation questions yields better hazard identification in health care", International Journal for Quality in Health Care, Vol. 31, No. 7, 2019, pp. 16-21.

[28] E. E. Odzaly, D. Greer, D. Stewart, "Agile risk management using software agents", Journal of Ambient Intelligence and Humanized Computing, Springer, Vol. 9, No. 23, 2018, pp. 823-841.

[29] A. Nadali, A. Grilo, A. Zutshi, "A conceptual framework of risk identification for scale up companies in transition period", Proceedings of the International Conference on Industrial Engineering and Operations Management, 2018, pp. 2346-2357.

[30] T. E. Abioye, O. T. Arogundade, S. Misra, A. T. Akinwale, O. J. Adeniran, "Toward ontology-based risk management framework for software projects: An empirical study", Journal of Software: Evolution and Process, Vol. 32, No. 12, 2020, p. e2269.

[31] M. Sousa, “Fast Brainstorm techniques with modern board games adaptations for daily uses in business and project managing”, Proceedings of the International Conference of Applied Business and Management, 2020, pp. 508-524.

[32] L. M. Khodeir, M. Nabawy, “Identifying key risks in infrastructure projects-Case study of Cairo Festival City project in Egypt”, Ain Shams Engineering Journal, Vol. 10, No. 3, 2019, pp. 613-621.

[33] M. Khalilzadeh, H. Shakeri, S. Zohrehvandi, “Risk identification and assessment with the fuzzy DEMATEL-ANP method in oil and gas projects under uncertainty”, Procedia Computer Science, Vol. 181, 2021, pp. 277-284.

[34] M. Zavvar, A. Yavari, S. M. Mirhassannia, M. R. Nehi, A. Yanpi, M. H. Zavvar, “Classification of risk in software development projects using support vector machine”, Journal of Telecommunication, Electronic and Computer Engineering, Vol. 9, No. 1, pp. 1-5.

[35] M. Yang, M. K. Lim, Y. Qu, X. Li, D. Ni, “Repair missing data to improve corporate credit risk prediction accuracy with multi-layer perceptron”, Soft Computing, Vol. 26, No. 18, 2022, pp. 9167-9178.

[36] B. Esmaeili, M. R. Hallowell, B. Rajagopalan, “Attribute-based safety risk assessment. II: Predicting safety outcomes using generalized linear models”, Journal of construction engineering and management, Vol. 141, No. 8, 2015, p. 04015022.

[37] Q. Wang, Y. Li, Y. Wang, J. Ren, “An automatic algorithm for software vulnerability classification based on CNN and GRU”, Multimedia Tools and Applications, Vol. 81, No. 5, 2022, pp. 7103-7124.