Educational Administration: Theory and Practice
2024,30(4), 1051-1057
ISSN:2148-2403
https://kuey.net/
Research Article

Predictive Analytics And Machine Learning For Real-Time Detection Of Software Defects And Agile Test Management

Md Fokrul Islam Khan[1]*, Abdul Kader Muhammad Masum[2]

[1]*M.Sc. in Management Information System International American University
Email:fokrulkhan837@gmail.com
ORCID ID: https://orcid.org/0009-0001-8174-4831
[2] Ph.D. Professor, Dept. of CSED affodil International University

Citation: Md Fokrul Islam Khan et Al. (2024), Predictive Analytics And Machine Learning For Real-Time Detection Of Software Defects And Agile Test Management...*Educational Administration: Theory And Practice*, *30*(4), 1051-1057
Doi: 10.53555/kuey.v30i4.1608

ARTICLE INFO

ABSTRACT

Supply chain agility is crucial for organizations striving to weather today's turbulent business climate. Agility requires proactive risk minimization. This paper gives a machine learning and predictive analytics strategy for real-time risk counteraction and dexterity. Conventional inventory network risk management utilizes post-occasion examination and authentic information, restricting its ability to deal with real-time interruptions. This paper proposes a modern methodology that utilizes predictive analytics to expect interruptions. Machine learning algorithms can identify patterns, correlations, and anomalies that indicate impending threats using contextual and historical data. By combining these models into a real-time monitoring system, organizations may detect and mitigate dangers. This research uses many predictive analytics methodologies to identify supply chain hazards. Natural language processing, time series analysis, and anomaly identification are examples. Risk assessment methods are also refined using machine learning techniques to ensure accuracy and flexibility. This study combines theoretical discourse with empirical data to explain the link between predictive analytics, machine learning, and supply chain agility.

Key Words : Predictive Analytics, Machine Learning, Real-Time Detection, Software Defects, Agile Test Management.

1. INTRODUCTION

Software engineering is the process of developing and using good engineering concepts to create dependable, reasonably priced software that works on actual hardware. A high-caliber software product requires a well-managed development process. The process of developing software is a human undertaking that entails several tasks. Each of the following activities—analysis, design, implementation, and testing—contributes to the production of the finished product. Because of these ongoing development operations, it may take some time to produce a functional version of the system. One of the main steps in software development is software testing, which is done to confirm and validate a software system. program developers may ascertain whether or not the detected issues have been resolved by testing to make sure the generated program performs as intended. The software testing process should be effective and efficient since the software development life-cycle is a complicated process that requires a new product to be delivered within the allotted period.

Software development and quality assurance are undergoing a revolution thanks to predictive analytics and machine learning, which make it possible to identify flaws in real time and improve agile test management techniques. These cutting-edge solutions provide major benefits in locating, ranking, and fixing bugs quickly in the quickly changing field of software engineering, where quality and agility are critical.
To anticipate future outcomes or patterns, predictive analytics utilizes factual calculations, verifiable information, and examples. When used in software development, it may examine enormous volumes of data gathered from several sources, including testing frameworks, bug tracking software, and version control systems, to predict possible flaws before they arise. Predictive analytics gives teams the ability to take

preventive steps by finding trends that point to flaws, which lowers the chance that important problems will surface throughout the development lifecycle.
Predictive analytics is enhanced by machine learning, which gives systems the ability to continuously learn from data and perform better over time without requiring explicit programming. Machine learning algorithms may be taught on labelled datasets including instances of known faults and non-defective code in the context of defect identification. The probability of faults is then predicted by these algorithms by comparing fresh code modifications or test results to previously identified trends. Machine learning models become better at spotting possible flaws and helping developers efficiently prioritize their work as more data becomes accessible.
Software defect detection in real time is essential to keeping agile development cycles moving forward. Defect discovery and resolution are often delayed by the use of human inspection or planned test runs in traditional testing procedures. Teams may monitor code modifications and testing operations continuously with the use of predictive analytics and machine learning, which can detect possible flaws early on and notify teams of them. By accelerating the feedback loop and enabling developers to address problems and iterate quickly, this proactive strategy helps developers provide higher-quality software in shorter amounts of time.

Agile test management techniques place a strong emphasis on cooperation, adaptation, and flexibility throughout across the software development lifecycle. Agile processes that use predictive analytics and machine learning provide teams with real-time information regarding the quality of their codebase and testing efforts. Agile teams may concentrate their attention on areas with the most risk or impact by combining automated defect detection with intelligent prioritizing processes. This maximizes productivity and reduces the amount of time it takes to resolve significant problems.

2. LITERATURE REVIEW

Zeineddine, H., Farah, A., and Braendle, U. (2021). Success for students has lately emerged as the main strategic goal for the majority of higher education institutions. Academic institutions are focusing more on keeping students enrolled in their programs without sacrificing rigor and quality of instruction due to budget constraints and rising operating expenses. Colleges are relying increasingly more upon information to estimate understudies' prosperity because of logical forward leaps in Huge Information Analytics and Machine Learning. The use of scholarly and social information from understudies to classify them and estimate their future exhibition utilizing complex insights and machine learning has been the subject of a few drives and examination projects. This exploration recommends using Robotized Machine Learning to work on the exactness of anticipating understudy execution utilizing information open before the beginning of the scholarly program, taking into account early mediation.

In 2020, Subbiah, U., Mahmood, Z., and Ramachandran, M. It is now unavoidable that a software release will have problems. When defects are present in a software release, a corporation might suffer enormous losses. The emphasis of contemporary testing and debugging techniques has moved from "detecting" to "predicting" faults in the code. The bug prediction models that are now in use have not been made commercially viable. Furthermore, there hasn't been a thorough discussion of these models' scalability yet. This chapter utilizes two ways to account for the different costs associated with resolving flaws based on the stage of the software development cycle the issue is identified in. One model may be used when the "cost of changing code" curve is exponential, while the other model can be used in other situations. We go over which scenarios each model works best in. This chapter offers software development organizations a model that may be implemented on a cloud platform. This chapter's model uses machine learning classification models to forecast if a problem in the code will exist or not. Bug prediction as a service (BPaaS) may be made available globally by the distribution of this model as a web service using Microsoft Azure's machine learning platform.

Mahdi, M. N., Yusoff, Y., Cheng, L. K., Ahmad, A. R., Mohamed Zabil, M. H., Ahmad, A. R., & Happala Naidu, H. (2021). Planning and evaluating project management is crucial to project performance activities. Project management is hard to do well without a reasonable and practical strategy. An exhaustive and complete outline of deals with the utilization of machine learning in software project management is introduced in this review. Furthermore, an exhaustive writing survey of three essential libraries — Web Science, Science Coordinates, and IEEE Investigate — as well as two segments on software project management and machine learning are remembered for this work. These three storehouses incorporate 111 papers coordinated into four classes. Exploration and review distributions on software project management are remembered for the principal class. Papers in light of machine-learning methodologies and strategies applied to projects fall into the subsequent classification; concentrates on the stages and tests that act as the boundaries for machine-learning management and the last classes of the review's outcomes, as well as review that add to the creation and advancement of machine-learning project expectation, are remembered for the third class. Furthermore, our commitment gives a structure and a more exhaustive perspective that may be important for future undertaking risk management work. At long last, we have exhibited that machine learning-based project risk evaluation is more compelling at diminishing venture misfortune, which raises the likelihood of task achievement. It likewise offers an alternate way to deal with really lower project disappointment probabilities, expanding the

result proportion for development, and making it more straightforward to break down software shortcoming expectation in light of exactness.
Linos, P. K., and S. Chenoweth (2023). An innovative undergraduate course at the nexus of machine learning (ML) and software engineering (SE) that is based on issues identified by the industry. According to ML experts, creating ML systems differs from creating software in such a way that we need fresh insights on how to integrate ML into software development. For example, a wide range of specialists, including statisticians, data scientists, and business analysts, must be closely engaged in these SE initiatives. the development of a table outlining and relating course themes, learning goals, and associated activities to industrial concerns. The course material was developed by surveys of undergraduate SE students and interviews with experts in the field who have relevant expertise. The intended course format is intended to mimic ML-based SE found in the real world. At the nexus of SE and ML, industry-derived material has been successfully created for a prototype undergraduate course.

Tatikonda, V. M., Vemuri, N., and Thaneeru, N. (2022). This study investigates how Deep Learning (DL) and DevOps approaches might be used to improve predictive maintenance in the manufacturing sector. Conventional maintenance approaches can result in operational interruptions, increased downtime, and inefficiency. Through the use of DevOps' agile concepts and the analytical powers of DL models, our research presents a complete framework for the early detection and mitigation of equipment failures. The materials and methods include preprocessing approaches to assure data quality along with data collecting from various sources, such as historical records and sensor data. Accurate equipment failure prediction is made possible by carefully choosing deep learning models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs). Consisting of automated testing, continuous deployment, and continuous integration, the integration pipeline adheres to DevOps concepts. The flexibility of the model to changing operating situations is ensured by real-time monitoring and feedback methods. System integration may be approached holistically when data scientists, software developers, and maintenance teams work together. Our approach paves the way for fast, effective, and adaptable predictive maintenance systems by addressing issues with cooperation, model drift, and security concerns. The combination of DevOps and DL is becoming more important as industrial sectors adopt digital transformation. This combination enhances asset dependability, promotes operational excellence, and helps manufacturing ecosystems develop sustainably.

Elmitwally, N. S., Khan, M. A., Ahmad, M., Fayaz, M., Abbas, S., Aftab, S., & Khan, F. (2022). The need for affordable, high-quality, and maintainable online software systems has grown along with the need for automated software systems. One of the most important parts of the quality assurance process is software defect prediction, which lowers costs by lowering the total amount of testing and maintenance required. The timely delivery of maintainable software, which eventually results in early repairs and satisfied customers who feel more trust in the development team, is a direct result of early defect identification in the software development life cycle (SDLC). Many machines learning-based methods have been presented in the last ten years to attain improved accuracy in software fault prediction. The majority of frameworks and models for defect prediction that have been developed employ artificial neural networks (ANNs), which are considered to be among the most popular machine learning approaches. This study offers a critical evaluation of the most recent research on the use of artificial neural networks for software defect prediction, which was published between 2015 and 2018. In this study, the literature is extracted using a methodical research technique from three popular digital libraries, including IEEE, Elsevier, and Springer. Eight of the most relevant research papers are then chosen for critical assessment after a rigorous selection process. By examining current developments in software defect prediction with an emphasis on ANNs, this study will benefit academics. It will also provide a baseline for future advances, comparisons, and reviews.

3. PROPOSED METHODOLOGY

3.1. Predictive Analytics in Supply Chain Risk Management

Predictive analytics, the groundwork of our proposed approach, utilizes both verifiable and present information to distinguish possible disturbances before they happen. Predictive analytics utilizes progressed measurable procedures along with machine learning calculations to recognize examples, connections, and patterns that conventional gamble management approaches could miss.

Data Collection and Processing

The research aimed to develop a real-time supply chain risk mitigation methodology by collecting data from both internal and external sources. Inward information included basic variables like lead times, stock levels, and past disturbance records, while outside sources included market developments, financial pointers, weather conditions conjectures, and international turns of events. Information preprocessing was critical for setting up the gathered information for examination, guaranteeing precision and consistency. Highlight designing was utilized to acquire noteworthy bits of knowledge and further develop forecast models. The reaction list included exact gamble elements and results to conjecture. These variables were painstakingly chosen to line up with research points and give significant experiences to take a chance with relief techniques. The dataset was thoroughly ready for additional investigation, guaranteeing the information's quality and significance were basic to the viability and legitimacy of the real-time production network risk relief approach.

Feature Engineering

Highlight designing is a vital stage in research, including the choice and control of huge factors to enter forecast models. This comprehends the unique idea of inventory network exercises and potential gamble triggers. The element assortment focuses on time-subordinate characteristics, context-oriented variables, and occasion pointers, representing transient examples, verifiable setting, outer market conditions, monetary information, and international occasions. By mindfully designing elements, the predictive force of the models is upgraded, empowering them to expect and relieve real-time inventory network gambles.

Model Selection and Training

This review utilizes time series investigation, relapse, and grouping calculations to foresee production network gambles. Time series examination is compelling in distinguishing and anticipating patterns in past information, considering foreseeing changes popular and production network breakdowns. Relapse investigation measures the effect of production network factors on execution. Characterization calculations sort and conjecture occasions or results, deciding the chance of risks or interruptions. The picked calculations are picked in view of their pertinence, demonstrated adequacy in past examinations, and similarity with contextual analyses. The review stresses the significance of staying aware of industry improvements and adjusting imagination and common sense. The calculations are prepared utilizing verifiable information and tested utilizing cross-approval techniques. Machine learning procedures are utilized to iteratively upgrade the viability of the models. To guarantee proceeded with precision, the review recommends a nonstop learning process, consolidating strategies like internet learning and moving time windows for refreshes. This approach guarantees the expectation models keep up with exactness and pertinence in the unique idea of contemporary stockpile chains.

3.2. Real-Time Monitoring and Risk Identification

The mix of predictive models into a powerful checking framework is vital for real-time assessment of information streams, including marketing projections, stock levels, lead times, and outside factors. The framework plans to identify oddities, blunders, or early admonition signs that could prompt store network interruptions or risks. It thinks about real-time information against anticipated models and cautions administrators when irregularities arise. This stage demonstrates the operationalization of predictive models, enhancing supply chain agility and resilience by enabling timely and informed responses to emerging hazards.

Anomaly Detection

Peculiarity detection strategies are vital in real-time following and hazard ID, distinguishing deviations or irregularities in information streams. These calculations dissect noticed values and patterns for boundaries like deals, stock levels, and lead times. They can distinguish expected early admonition indications of issues, for example, startling deals increments or store network issues. Peculiarity detection assists associations with making a move, for example, changing creation plans, reallocating assets, or tracking down new providers, to keep up with functional progression.

3.3. Proactive Risk Mitigation and Adaptive Strategies

At the point when potential perils are perceived, the structure advances the turn of events and execution of proactive gamble alleviation methodologies. These techniques utilize past information, machine learning experiences, and pre-laid out reaction systems to ensure a powerful and proficient dynamic cycle.

Risk Impact Assessment

Predictive analytics makes it more straightforward to measure possible impacts of recognized dangers on store network tasks. This evaluation advises the prioritization regarding reaction measures and the conveyance of assets. During the gamble influence assessment stage, the conceivable effect of distinguished takes a chance on store network tasks is estimated. This assessment aids in prioritizing the relevant reaction activities based on the severity of the hazards. The effect evaluation can be quantified by combining elements and characteristics relevant to the specific supply chain environment.

Adaptive Decision-Making

The system uses machine learning algorithms to continuously improve risk assessment models, ensuring its versatility even with changing risk profiles. To validate the framework's effectiveness, researchers use a quantitative methodology, comparing its performance indicators with actual results. This helps measure the alignment between anticipated risks and actual disruptions, and determine its contribution to mitigating disruptions. The quantitative approval approach likewise surveys the functional and monetary consequences of the system suggested exercises, empowering correlations and showing its advantages in further developing production network deftness and versatility

4. RESULT AND DISCUSSION

The consequences of the execution should be considered while evaluating the proposed structure for production network risk management's coordination of machine learning and predictive analytics. The framework's ability to increase supply chain resilience and agility was extensively evaluated using a number of well-thought-out procedures and data-driven research projects. This part offers a comprehensive analysis of the empirical results obtained from applying the framework to both simulated and real-world environments. The results provide insight into the framework's ability to recognize threats, anticipate hiccups, assess potential outcomes, and provide well-informed solutions. Additionally, they illustrate the tangible results of every step within the framework. The quantitative results provide valuable insight into the benefits and drawbacks of the framework and provide a clear picture of how it supports supply chain decision-making and risk reduction.

Table 1: Predictive Analytics Results

Model used	Training data size	Testing data size	Accuracy%	Precision%	Recall%	F1-score %
One– class SVM	8000	2000	89.5	83.1	76.4	79.7

Table 2: Early Warning System Results

Alert threshold	True positives	False positives	True negative	False negative	Precision%	Recall%	F1-score %
0.75	147	34	1652	36	80.8	82.6	83.4

An early warning system's outcomes are shown in Table 5, which provides information on how well it performed at various alarm levels. The confidence levels at which the system signals possible problems or anomalies are represented by these thresholds. Important data including true positives, false positives, true negatives, and false negatives are included in the table. These parameters add up to a confusion matrix that helps assess how well the system detects abnormalities. The algorithm properly discovers 147 true positives, or anomalies, at an alert level of 0.75, but generates 34 false positives, or typical occurrences that are mistakenly marked as anomalies. Furthermore, it accurately detects 1652 real negatives, or cases that are normal, but fails to detect 36 abnormalities, leading to false negatives. Metrics including precision, recall, and F1-score are used to evaluate the system's effectiveness. The system shows an accuracy of 80.8% at the same threshold, meaning that around 80.8% of anomalies that are identified are real. Its 82.6% recall, or sensitivity, indicates that around 82.6% of real abnormalities are found. The system's total performance in detecting anomalies, taking into account false positives and false negatives, is shown in the F1-score, which is 83.4% and combines accuracy and recall. These findings provide insightful information about the system's advantages and shortcomings, especially with regard to lowering false positives and false negatives to raise accuracy and recall rates.

Table 3: Risk Impact Assessment Results

Risk type	Impact level	Revenue impact % reduction
Supply shortage	High	14%
Demand surge	Moderate	10%
Geopolitical risk	High	16%
Supplier delay	Moderate	12%
Regulatory change	Low	6%
Market shift	Moderate	8%
Supplier insolvency	High	17%
Production failure	High	16%

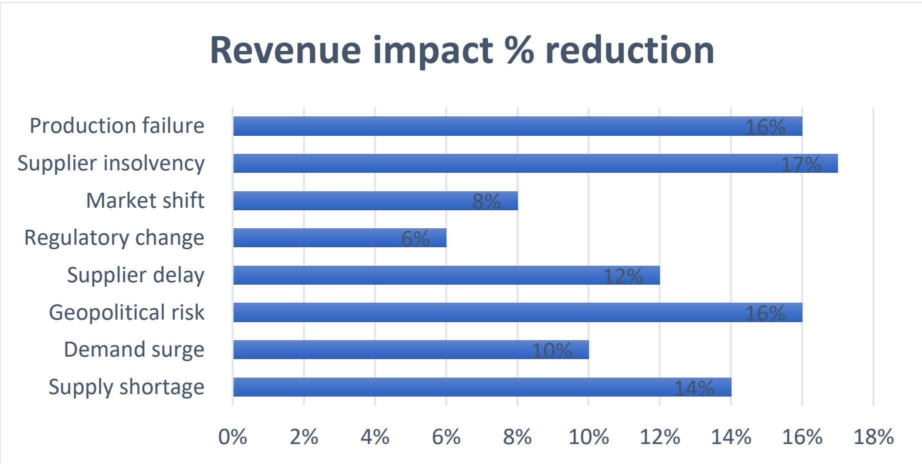

Figure 1: Graphical Representation of Revenue Impact % Reduction

The accompanying table provides information on the wide range of risks that an organization may face, classifying them according to their nature, degree of effect, and the percentage of revenue impact that may be reduced by mitigation measures. It includes a variety of hazards, such as shifts in regulations and geopolitical instability, as well as supply shortages and demand spikes. Every risk category is rated from high to low according to how much of an influence it has. Operations and income streams are seriously threatened by high impact risks, which include manufacturing failures, supplier bankruptcy, geopolitical hazards, and shortages of supplies. However, with reductions ranging from 14% to 17%, mitigation actions have shown to be extremely successful in lessening their influence on income. While not as important as high impact risks, moderate effect risks still need to be taken into consideration. These include changes in the market, supplier delays, and spikes in demand. However, mitigating strategies have resulted in significant drops in revenue effect, from 8% to 12%. Consideration should be given to even low impact risks, such as regulatory changes, as they have the potential to harm income. Mitigation actions have reduced the effect on revenue by 6%, notwithstanding their comparatively modest impact. The chart, taken as a whole, emphasizes the significance of thorough risk management plans that are customized to the unique characteristics and degree of each risk category. Organizations may improve their resilience to possible disruptions and protect their income by allocating mitigation efforts in a manner that makes sense.

Table 4: Adaptive Decision-Making Results

Model update interval	Improved response time %	Resource allocation efficiency %
6 h	25%	19
12 h	17%	16
24 h	13%	10
48 h	9%	8

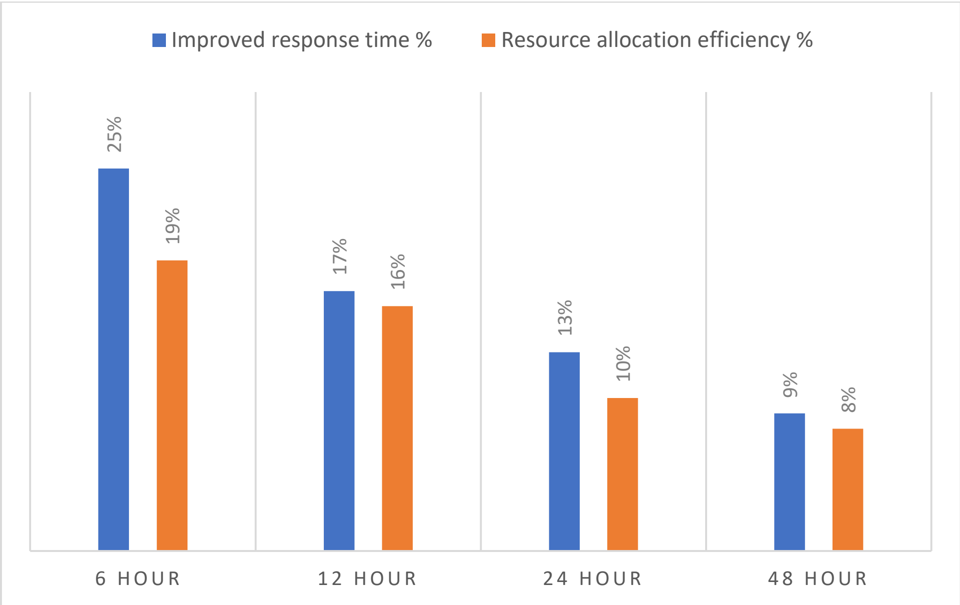

Figure 2: Graphical Representation Of Adaptive Decision-Making Results

The results of adaptive decision-making techniques based on different intervals for updating the decision-making model are shown in the table. The model may be updated at intervals of six hours, forty-eight hours, or fifty hours. The percentage improvement in reaction time and the percentage increase in the efficiency of resource allocation are the two main criteria that are evaluated. The data shows a distinct trend: improved performance across both measures is the outcome of more frequent modifications to the decision-making model. In particular, compared to less frequent updates, refreshing the model every 6 hours results in the largest gains in reaction time, with a 25% increase. In a similar vein, this period exhibits the most improvement in resource allocation efficiency, up 19%. But the improvement decreases in size as the update interval grows. For example, updating the model every 48 hours results in the least amount of improvement, just 9% and 8%, respectively, in reaction time and resource allocation efficiency. This emphasizes how crucial it is to update the model often in order to improve resource allocation efficiency and reaction times in adaptive decision-making processes. However, companies need to weigh the advantages of regular upgrades against the necessary infrastructure and processing resources.

5. CONCLUSION

We utilized predictive analytics and machine learning to make a clever inventory network risk management strategy. Our review demonstrates the way that this innovation can upset production network deftness. By real-time risk moderation, associations might guarantee functional coherence, streamline asset portion, and lift consumer loyalty. As found for our situation studies, this procedure demonstrated viable in various areas, exhibiting its flexibility to request vacillation and international unusualness. Looking forward, the changing store network management situation has provoked a few exploration headings. To ensure real-time predictive model viability, algorithm complexity and computing efficiency must be balanced. Attempts to simplify machine learning algorithms may boost supply chain specialists' confidence. Researching ways of incorporating blockchain and IoT into our framework might build its constancy and receptiveness. Moreover, understanding hierarchical elements and social changes required for viable reception, as well as human dynamic utilizing predictive analytics and machine learning, are fundamental exploration fields.

REFERENCES

1. Ahmed, M. R., Ali, M. A., Ahmed, N., Zamal, M. F. B., & Shamrat, F. J. M. (2020, January). The impact of software fault prediction in real-world application: an automated approach for software engineering. In Proceedings of 2020 6th International Conference on Computing and Data Engineering (pp. 247-251).
2. Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., ... & Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. Automation in Construction, 141, 104440.
3. Biesialska, K., Franch, X., & Muntés-Mulero, V. (2021). Big Data analytics in Agile software development: A systematic mapping study. Information and Software Technology, 132, 106448.
4. Chenoweth, S., & Linos, P. K. (2023). Teaching Machine Learning as Part of Agile Software Engineering. IEEE Transactions on Education.
5. Devale, P., Jadhav, R. B., Bidwe, R. V., Mane, M., & Pawar, P. (2024). Machine Learning and Just-in-Time Strategies for Effective Bug Tracking in Software Development. International Journal of Intelligent Systems and Applications in Engineering, 12(6s), 749-758.
6. Hanslo, R., & Tanner, M. (2020, September). Machine learning models to predict agile methodology adoption. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 697-704). IEEE.
7. Kabanda, G. (2021). Performance of machine learning and big data analytics paradigms in cybersecurity and cloud computing platforms. Global Journal of Computer Science and Technology, 21(G2), 1-25.
8. Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps practices of continuous automation for machine learning. Information, 11(7), 363.
9. Khan, M. A., Elmitwally, N. S., Abbas, S., Aftab, S., Ahmad, M., Fayaz, M., & Khan, F. (2022). Software defect prediction using artificial neural networks: A systematic literature review. Scientific Programming, 2022.
10. Mahdi, M. N., Mohamed Zabil, M. H., Ahmad, A. R., Ismail, R., Yusoff, Y., Cheng, L. K., ... & Happala Naidu, H. (2021). Software project management using machine learning technique—A Review. Applied Sciences, 11(11), 5183.
11. Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193.
12. Subbiah, U., Ramachandran, M., & Mahmood, Z. (2020). Software engineering framework for software defect management using machine learning techniques with azure. Software Engineering in the Era of Cloud Computing, 155-183.
13. Vemuri, N., Tatikonda, V. M., & Thaneeru, N. (2022). Integrating Deep Learning with DevOps for Enhanced Predictive Maintenance in the Manufacturing Industry. Tuijin Jishu/Journal of Propulsion Technology, 43(4), 2022.]
14. Woschank, M., Rauch, E., & Zsifkovits, H. (2020). A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics. Sustainability, 12(9), 3760.
15. Zeineddine, H., Braendle, U., & Farah, A. (2021). Enhancing prediction of student success: Automated machine learning approach. Computers & Electrical Engineering, 89, 106903.