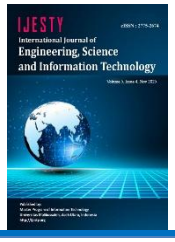
International Journal of Engineering, Science and Information Technology
Volume 5, No. 3 (2025) pp. 595-602
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.1370
Research Paper, Short Communication, Review, Technical Paper



Safety Function Model for Requirement Specification in Critical Systems: A Case Study of Generic Patient Controlled Analgesia Pump Model

Azma Abdullah[1]*, Rohani Abu Bakar[1], Fairus Abdul Farid[2], Mansoor Abdulhak[3]

[1]Department of Software Engineering, Faculty of Computing, University Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia
[2]Malaysia Nuclear Agency, Selangor, Malaysia
[3]School of Computer Science, Gallogly College of Engineering, University of Oklahoma, United States

**Corresponding author Email: azma@umpsa.edu.my*

The manuscript was received on 25 January 2025, revised on 17 June 2025, and accepted on 22 July 2025, date of publication 26 July 2025

Abstract

Developing safety-critical systems (SCS) involves a systematic method for assuring and providing safety and dependability. Conventional approaches rely on expert intervention, which can introduce bias, cause delays, and promote inconsistency. This work proposes a model that enhances efficiency and accuracy by extracting safety functions from requirements specifications. The model is made up of three main steps: (1) preprocessing, which involves getting rid of stop words; (2) string selection and matching using a database of safety properties variables based on literature and expert knowledge; and (3) putting safety and non-safety functions into a structured safety function log. The model was trained and tested with the CGPA insulin pump and got a 94% F1 measure score, which means it was 91% accurate, 96% accurate, 92% precise, and 96% recall. This shows that it is good at making things clearer and less biased when finding functions for safety against failures, malfunctions, operational hazards, and inconsistencies in safety-critical specifications. All these enhancements contribute towards Sustainable Development Goal (SDG) 11: Sustainable Cities and Communities, aiming to develop safer, resilient, and sustainable infrastructure in safety-critical regions.

Keywords: *Safety Function, Safety Function Model, Requirement Specification, Safety Critical Systems.*

1. Introduction

Safety-critical systems (SCSs) are those whose failure can have disastrous consequences, including loss of life, environmental degradation, and financial loss [1]–[3] SCSs run in numerous industries [4], including medical care, airlines, and nuclear power, in which dependability is a significant issue. As an instance, the CereLink ICP Monitor, a crucial device for regulating intracranial pressure in critical care, was recalled by the U.S. Food and Drug Administration (FDA) due to its inaccurate blood pressure readings. This device poses a high risk of infection, bleeding, and tissue trauma, as evidenced by at least 105 worldwide complaints and 68 Medical Device Reports (MDRs) received until August 24, 2022 [5]. Identifying safety functions in system development entails a range of roles, including domain experts, who validate and verify safety requirements [6]–[8] Domain expert use, however, involves some limitations [9]–[11], including expertise bias [12]–[14], time constraints [14]–[16], and issues of transparency [8], [10]. Researchers have explored the use of safety function models as an alternate method for identifying and managing safety requirements.

Recent research identifies a persistent challenge in specifying requirements and defining safety functions for safety-critical systems. Martins & Gorschek [17] report a need for enhanced elicitation, analysis, and verification techniques for safety requirements, elicited through practitioner interviews in a range of industries. On the other hand, Wu et al. [18] introduce functional modeling as a tool for describing system solutions and identifying failures, but its application in safety and risk analysis is limited. Furthermore, Hendrix et al. [19] introduce model-based approaches for increased accuracy and verification of system safety analysis, with a specific consideration for complex software-intensive systems, providing a systemic model for identifying and minimizing potential danger. Nouri et al. [20] then investigate the use of large language models in automating the refinement and decomposition of requirements, specifically in the automotive industry, whose constant updating generates a recurring challenge. Meanwhile, Chen et al. [9] introduce a formal technique for increased efficiency through arming domain professionals with tools for verification, minimizing lengthy dialogue with formal

professionals. Nevertheless, in consideration of such improvements, expert consultation continues to have a key role in accuracy and uniformity in safety requirement verification. Ultimately, overcoming such a challenge necessitates an acceptance of the key role played by expertise in specifying safety functions and developing safety requirements. AI-powered automation, model-based techniques, and functional modeling facilitate safety and efficiency in systems analysis. However, their effectiveness depends on closing knowledge gaps, with extensive collaboration between experts and systematic safety methods for ensuring rigorous requirements verification and safety confirmation for complex systems.

Building on the challenges of automating safety function modelling in requirement specifications, this paper proposes a safety model that uses a safety properties variables database that integrates and archives expert domain knowledge in a central repository, reducing the direct workload for experts while ensuring compliance with all regulatory requirements. The proposed model involves three phases: in the first, it removes any irrelevant words in the specification of a requirement; in the second, it analyzes terms via preprocessing to extract useful strings; and in the third, it creates a log of a document's safety function and discerns contrast between function types of both safety and non-safety. In a case study for a general patient-controlled analgesia pump, the result in comparing accuracy, precision, recall, and F-measures is examined. Recent studies have increasingly underpinned the imperative for well-designed databases of safety properties variables in a range of industries. Sheehan et al. [21] examined publicly available databases of patient safety and determined that most have not incorporated key capabilities for effective analysis for safety professionals and for data scientists, and consequently have little actual utility in practice. Similarly, Gupta et al. [22] emphasized industrially the imperative for a general-purpose database of values for thermodynamic and transport properties, a requirement for safe and reliable development of chemical processing. Projecting onto fire safety, McKinnon & Bellamy [23] documented the Fire Safety Research Institute Materials and Products database, providing material property and fire testing information for use in reliable modeling of fires. Together, previous studies make a strong case for a critical imperative for general-purpose, accessible, and well-designed databases of safety properties variables for enhancing analysis, verification, and modeling in a range of industries.

Safety properties variables also make a major contribution towards requirements engineering and requirements analysis of safety-critical systems, to achieve dependability and prevention of failures. Describing such properties in requirements specifications, Maurya & Kumar [24] note, it adds to the greater dependability of a system. Following the same direction, V. Nguyen Tran et al. [25] present integrated requirements engineering for software safety for resisting software failure in a safety-critical environment. Jensen and Tumer [26] present a "safety function" for identifying a system's safety property, integrating with performance functions, and providing for critical events for investigation. Following a complementary direction, Hamidi et al. [27] present a model for safety-related function measurement in terms of probability, with architectural decisions, and comparing safety and availability measurements in view. Thus, these works substantiate the major contribution of safety properties variables towards greater resilience of a system and resisting risk in a variety of safety-critical environments.

Furthermore, this paper has been focusing on safety function identification in requirements specification. The paper has been structured such that Section 2 explains the model's development, Section 3 presents the experimental evaluation, and Section 4 concludes the paper and suggests future research.

2. Methods

The development of the proposed model is represented in Figure 1. Figure 1 shows a methodology for developing a safety function model for requirement specification in safety-critical systems. The activity is initiated with a problem statement, scope, and objectives determination through a review of the literature. Next, a development of a safety properties variables database is derived through expert domain expertise and review of the literature, and it will be discussed in the following section. The database is then used in the safety function model process. The model is subjected to thorough testing, evaluation, and analysis for its effectiveness. If the output is acceptance, then the activity ends; else, a refinement of the model and re-evaluation is conducted till an acceptable output is achieved. That iterative manner ensures developing a strong and reliable Safety Function Model for requirements specification.

Fig 1. Research Methodology

2.1. Development of Safety Properties Variable Database

Figure 2 portrays a process for developing a variable database for safety properties variables using a combination of SLR and expert domain knowledge. SLR identifies the variable for safety properties through a review of scientific, peer-reviewed, and published studies with high academic standards. SLR results have been presented in [28]. By mixing both approaches, deeper, reliable, and scientific integrity in the derived safety properties variables databases is attained. On the one hand, expert domain expertise can also be utilized to determine safety property variables through professionals' field experiences and expertise. By combining expert estimation with data techniques, a larger and richer pool of safety property variables is constructed.

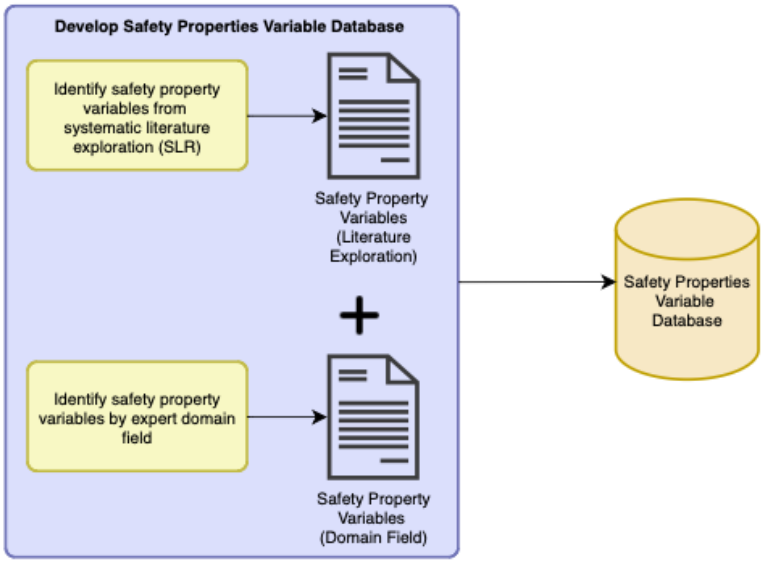


Fig 2. Develop Safety Properties Variables Database

Expert knowledge plays a significant role in discovering and confirming safety properties in safety-critical domains. Fazl Barez et al. [29] utilized expert knowledge in first-order logic for its application in reinforcement learning, providing safer exploration and increased efficiency in samples. In a similar direction, Xiaohong Chen et al. [9] developed SafeNL, a system that helps expert professionals in a specific field verify compliance with requirements for safety through formality and ease of collaboration with formal professionals. In materials science, Yue Liu et al. [30] developed the DML-FSdek method, which employs expert knowledge in weighted scoring for feature selection in predicting properties in materials, with an improvement in feature selection for property prediction in materials. For critical system evaluation of safety, Ievgen Babeshko et al. [31] developed XMECA and EUMECA techniques, with a combination of expert judgments and uncertainty for increased dependability in safety analysis. Together, these techniques present a significant role for expert knowledge in increased efficiency, compliance, and trust in security evaluation in a range of industries.

2.2. Safety Function Model

Figure 3 shows how the process of separating between safety and non-safety functions is performed in the proposed safety function model. Step 1: Stop word removal is initiated with processing, filtering out unnecessary words. Stop word removal adds accuracy because irrelevant terms are not considered, and this creates a cleaner keyword search. This adds safety property variable analysis and classifies safety functions more effectively.

In Step 2, the requirements specification is processed to extract information pertaining to safety. First, a relevant field of a domain is selected to convey contextual pertinency for requirements analysis for safety. In Step 2.2, the processed requirements text is then compared with a safety properties variables database. In this process, each requirement will be carefully checked to see if its key term aligns with predefined safety property variables. In case a key term is detected, a requirement is classified as a safety function; otherwise, it is considered a non-safety function. If a match is found (YES), then a safety function is documented in Step 3: Safety Function Log for further checking and documentation. In case no match is found (NO), then a non-safety function is stored in a file. The output of these steps is a list of found safety functions and non-safety functions.

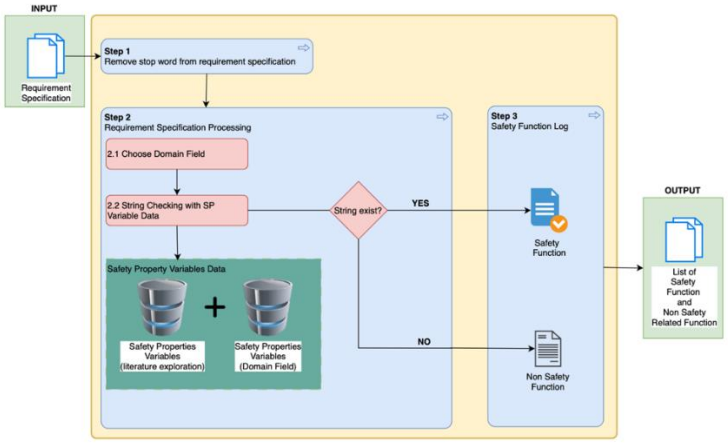


Fig 3. Proposed Safety Function Model

2.3. Evaluation of Evaluates Purpose Model

This model uses a case study and a structured experimental setup. This makes sure that the model is thoroughly evaluated for how well it finds and groups safety property variables according to the requirements.

2.3.1. Case Study

Figure 4 illustrates a GPCA system in a hospital setting for the administration of drugs [32][33]. The patient orders a bolus, processed via GPCA, delivering a flow of drugs through a needle. Prescription information and infusion orders entered by the clinician receive notification information from GPCA. GPCA talks to the Hospital Pharmacy Database, ensuring the security of drugs by checking for security information of drugs and the Drug Reservoir supplying drugs in demand. Information flows (broken lines) represent computer communications, and direct delivery (full lines) represents interfaces in a physical form. All these provisions enable safe and computerized infusions with less opportunity for human errors. A partial view of the requirement specification for CGPA is shown in Figure 5(a).

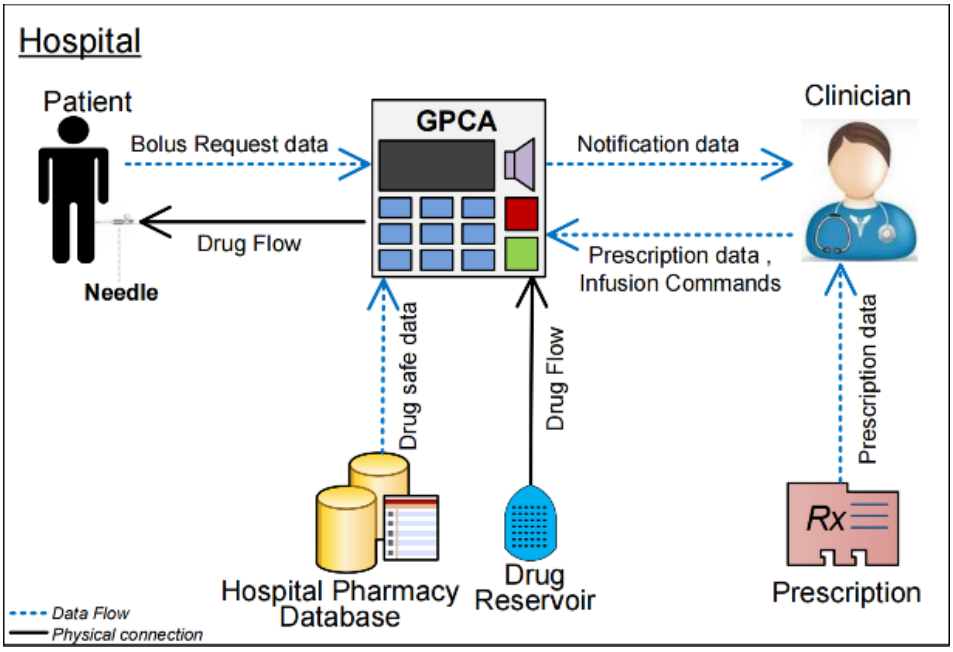


Fig. 4. GPCA Infusion System [32]

To be utilized in the model, the requirement specification was converted into a formatted list and prepared as an input file, as illustrated in 5 (b).

The GPCA Model ver. 0.9

FOR RESEARCH PURPOSES ONLY

Safety Requirements for the Generic Patient Controlled Analgesia Pump

This document lists safety requirements for the generic patient controlled analgesia (GPCA) pump model. The requirements include safety features and constraints for the GPCA pump. Configuration parameters for the model are identified and enumerated based on these requirements. All values (indicated in red) are parameters that can be configured based on specific implementations or extensions of the GPCA.

1. **Infusion Control**
   - 1.1. Flow rate
     - 1.1.1. The flow rate for the pump (for both primary and secondary infusions) shall be programmable.
     - 1.1.2. At minimum, the pump shall be able to deliver primary (basal) infusion at flows throughout the range of f_{min} to f_{max} *ml/hr*.
     - 1.1.3. Flow discontinuity at low flows (*f ml/hr or less*) should be minimal.
     - 1.1.4. The basal delivery rate shall be programmable for durations of up to *t hours*.
     - 1.1.5. An active basal shall continue to be delivered without change while programming basal rates.
     - 1.1.6. The pump should maintain a minimum KVO (keep vein open) rate of *x ml/hr* at all times during infusion.
   - 1.2. Flow rate accuracy
     - 1.2.1. During extended operation, the flow rate shall remain accurate within *±n%* of the rate setting for at least *t hours* of continuous use.
     - 1.2.2. If the pump is equipped with a flow rate sensor and the flow rate exceeds the programmed rate setting by more than *n%* over a period of more than *t minutes*, or if the pump goes into free flow, the pump shall issue an alarm to indicate overinfusion of the patient.
     - 1.2.3. If the pump is equipped with a flow rate sensor and the flow rate is less than *n%* of the programmed rate setting over a period of *t minutes*, the pump shall issue an alarm to indicate underinfusion of the patient.
   - 1.3. Volume to be infused (VTBI)
     - 1.3.1. The VTBI (Volume to be Infused) settings shall cover the range from v_{min} to v_{max} *ml*.
     - 1.3.2. The user shall be able to set the VTBI in *j ml* increments for volumes below *x ml*.
     - 1.3.3. The user shall be able to set the VTBI in *k ml* increments for volumes above *x ml*.

(a)

1. Infusion Control, Flow rate, The flow rate for the pump (for both primary and secondary infusions) shall be programmable
2. Infusion Control, Flow rate, At minimum, the pump shall be able to deliver primary (basal) infusion at flows throughout the range of f_{min} to f_{max} *ml/hr*
3. Infusion Control, Flow rate, Flow discontinuity at low flows (*f ml/hr or less*) should be minimal.
4. Infusion Control, Flow rate, shall be programmable for durations of up to *t hours*.
5. Infusion Control, Flow rate, An active basal shall continue to be delivered without change while programming basal rates.
6. Infusion Control, Flow rate, The pump should maintain a minimum KVO (keep vein open) rate of *x ml/hr* at all times during infusion.
7. Infusion Control, Flow rate accuracy, during extended operation, the flow rate shall remain accurate within *±n%* of the rate setting for at least *t hours* of continuous use.
8. Infusion Control, Flow rate accuracy, if the pump is equipped with a flow rate sensor and the flow rate exceeds the programmed rate setting by more than *n%* over a period of more than *t minutes*, or if the pump goes into free flow, the pump shall issue an alarm to indicate over infusion of the patient.
9. Infusion Control, Flow rate accuracy, if the pump is equipped with a flow rate sensor and the flow rate is less than *n%* of the programmed rate setting over a period of *t minutes*, the pump shall issue an alarm to indicate under infusion of the patient.
10. Infusion Control, Volume to be infused (VTBI), the VTBI (Volume to be Infused) settings shall cover the range from v_{min} to v_{max} *ml*.
11. Infusion Control, Volume to be infused (VTBI), the user shall be able to set the VTBI in *j ml* increments for volumes below *x ml*.
12. Infusion Control, Volume to be infused (VTBI), the user shall be able to set the VTBI in *k ml* increments for volumes above *x ml*.

(b)

Fig 5. (a) Safety Requirement Specification from Sources [31] and (b) Requirement Specification used for Proposed Model

2.3.2. Experimental Setup

Table 1 presents a collection of 107 functional requirements, representing ground truth, out of which 87 have been labelled as Safety Functions (SF) and 20 have been labelled as Non-Safety Functions (Non-SF). For model performance evaluation, its training and testing sets have been taken out of a split in the dataset. There are 75 functional requirements (61 SF and 14 Non-SF) in its training set and 32 functional requirements (26 SF and 6 Non-SF) in its testing set, which represent 70% -30%.

Table 1. Training and Testing Requirement Specification

Item	**Safety Function (SF)**	**Non-Safety Function (Non-SF)**
Ground truth data (107 functional requirements)	87	20
Data for training (75 functional requirements)	61	14
Data for testing (32 functional requirements)	26	6

3. Results and Discussion

3.1. Evaluation Phase

The evaluation section compares testing and training performance in an evaluation of model performance. To assess the performance of the proposed modeling requirements, the confusion matrix is employed. In a confusion matrix, a critical analysis of model performance is conducted through a count of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) values. In these experiments, two (2) different scenarios were considered, which are the testing of the model without expert knowledge while developing the safety properties variables, and another scenario is with expert knowledge. Performance values such as precision, recall, and F1-score, which are important in the evaluation and determination of improvement areas, are calculated through a calculation in a confusion matrix, and these have been represented in Equations 1 through 4 below.

$$Accuracy = \sum_{i=1}^{N} \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i} \quad (1)$$

$$Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FP_i)} \quad (2)$$

$$Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i} \quad (3)$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

3.2. Result for Training Model

The ground truth training data contains 61 Safety-Related Functions (SF) and 14 Non-Safety-Related Functions (Non-SF). Two scenarios apply: one without a field domain insulin safety properties (SP) variables database, and another with a safety property variables database. Model training performance is displayed in Table 2, comparing with and without expert contribution in developing a safety property database. With no expert contribution, 41 cases of safety functions were detected, with 34 correct detections (TP), 7 incorrect detections, and 27 missed cases (FN). For non-safety cases, 34 cases were detected, with 7 correct detections (TN), 7 incorrect detections (FP), and 27 incorrect detections (FN). With expert contribution, model performance increased, with 70 detected safety functions, 59 correct detections, 2 missed safety functions, and 11 incorrect detections. For the non-safety function, 5 requirements were detected, with 3 correct detections but 11 incorrect detections (FP) and 2 incorrect detections (FN). Performance reveals that expert contribution raises accuracy in function detections, with fewer requirements of incorrect detections and high confidence in classification.

Table 2. Proposed Model Training Result

Item	Safety Function (61 Requirements)				Non-Safety Function (14 Requirements)			
	Model Detect	Model Correct Detect (TP)	Model Missed Detect (FN)	Model Uncorrect Detect	Model Detect	Model Correct Detect (TN)	Model Missed Detect (FP)	Model Uncorrect Detect
Without Field Domain Insulin SP Variables Database	41	34	27	7	34	7	7	27
With Field Domain Insulin SP Variables Database	70	59	2	11	5	3	11	2

Table 3 compares performance with and without training with field domain insulin SP variable data and sums them below. Without training with field domain insulin SP variable data, when trained, it reached an accuracy of 0.55, a precision of 0.83, recall of 0.56, and an F1-score of 0.68. With training with field domain insulin SP variable data, its performance increased significantly, reaching 0.83 accuracy, 0.84 precision, 0.97 recall, and an F1-score of 0.90. The performance confirms that with field domain insulin SP variable data, its accuracy in predicting correct safety functions is enhanced, and recall and overall classification accuracy are enhanced.

Table 3. Summary Result for Training Data

	Without Field Domain Insulin SP Variable Data	With Field Domain Insulin SP Variable Data
Accuracy	0.55	0.83
Precision	0.83	0.84
Recall	0.56	0.97
F1–score measure	0.68	0.90

3.3. Result for Testing Model

Tables 4 and 5 present model testing and performance evaluation considering the insulin SP variable database. There are 26 Safety-Related Functions (SFs) and 6 Non-Safety-Related Functions (Non-SFs) from the ground data. The experiment results show there were 26 detected safe functions with 24 correct identifications (true positive, TP), 2 missed detections (FN), and 2 incorrect identifications from the model. For non-safe functions, 6 detections with 5 correct identifications (true negatives, TN), 1 misidentification (false positive, FP), and 1 incorrect detection were made. All these statistics validate that field domain insulin SP variables integration strengthens model performance in the correct classification of safe functions, with reduced incorrect detections and missed critical safe items.

Table 4. Model Testing Result

Item	Safety Function				Non Safety Function			
	Model Detect	Model Correct Detect (TP)	Model Missed Detect (FN)	Model Incorrect Detect	Model Detect	Model Correct Detect (TN)	Model Missed Detect (FP)	Model Incorrect Detect
With Field Domain Insulin SP Variable Database	26	24	2	2	6	5	1	1

Table 5 is a model testing performance summary, with model accuracy at 0.91, indicative of strong overall performance in terms of classification. Precision value 0.92 is indicative of model performance in terms of minimizing incorrect positive cases, and recall value 0.96 is indicative of its performance in terms of finding most actual safety functions with fewer incorrect negatives. An F1-score value of 0.94 is indicative of a balanced reconciliation between recall and precision, indicative of model robustness. All these values denote a considerable improvement in performance in terms of classification with field domain insulin SP variables, indicative of its effectiveness in terms of improving model dependability in finding requirements critical for safety.

Table 5. Summary Result for Testing Data

Measurement	Insulin SP Variable Data
Accuracy	0.91
Precision	0.92
Recall	0.96
F1–score measure	0.94

Table 6 summarizes the training and testing performance for the model. In training, 0.83 accuracy, 0.84 precision, 0.97 recall, and 0.90 F1-score represent high performance in the function for the determination of safety. In testing, performance continued with 0.91 accuracy, 0.92 precision, 0.96 recall, and 0.94 F1-score, proving dependability in function classification for the determination of safety in new, unseen information. Overall, model generalizability between training and testing can be regarded as high, with high recall and precision for the correct determination of the function of safety.

Table 6. Summary Training and Testing Results for Model Performance

	Training Result	**Testing Result**
Accuracy	0.83	0.91
Precision	0.84	0.92
Recall	0.97	0.96
F1–score measure	0.90	0.94

The results indicate that the use of expert input and field-domain insulin SP variable data significantly enhances safety function detection accuracy and reliability. The model has fewer errors of incorrect classification, higher levels of confidence, and higher recall and precision, proving its reliability in safety-critical requirements identification. The use of field-specific variables also boosts model performance, with safety function classification and minimal omitted critical safety items. The findings suggest significant classification accuracy improvement, proving the efficacy of the model in safety function extractions from requirements specifications. The model also exhibits high generalizability between training and testing, proving its resilience for real-world use.

4. Conclusion

For safety-critical systems (SCS), a model is developed for automating the classification of requirements specifications for safety functions, in contrast with conventional expert-only practice. In its model, both expert and literature requirements specifications for a domain are included to improve accuracy in classification. Once processed, the model performs all operations sequentially until a

conclusive classification. For its performance evaluation, a public dataset named the CGPA insulin pump has been tested. On its datasets, the proposed model was executed for generating its conclusive classification output.
The results achieved are 91% accuracy, 92% precision, 96% recall, and an F1 score of 94%, proving effectiveness in function classification for those pertaining to safety-related ones. The model proposed can confirm that it can serve experts through automation, improving efficiency and processing high requirements specifications in seconds, according to this work. Consequently, the model can serve as a semi-automated tool in supporting domain experts in distinguishing between non-safety and safety requirements. Subsequently, including Explainable AI (XAI) in future work can enhance transparency in safety-critical situations, with experts being able to know and trust AI-powered. Although useful in terms of highlighting, AI-powered knowledge bases, dynamically developed, updated, and about safety-related function solutions, will make the model adaptable to the changing requirements about safety. All these enhancements will drive Sustainable Development Goal (SDG) 11 through developing safer, more resilient, and sustainable ones pertaining to safety-critical ones.

Acknowledgement

This research was funded by a grant from Universiti Malaysia Pahang Al-Sultan Abdullah (RDU210313).

References

[1] K. Hobbs, M. Mote, M. Abate, S. Coogan, and E. Feron, "Run Time Assurance for Safety-Critical Systems: An Introduction to Safety Filtering Approaches for Complex Control Systems," *IEEE Control Syst.*, vol. 43, no. 2, pp. 28–65, Jun. 2022, doi: 10.1109/MCS.2023.3234380.
[2] L. Buysse, I. Habli, D. Vanoost, and D. Pissoort, "Safe autonomous systems in a changing world: Operationalising dynamic safety cases," *Saf. Sci.*, vol. 191, p. 106965, Nov. 2025, doi: 10.1016/J.SSCI.2025.106965.
[3] X. Wang, J. Yang, C. Liu, Y. Yan, and S. Li, "Safety-Critical Disturbance Rejection Control of Nonlinear Systems With Unmatched Disturbances," *IEEE Trans. Automat. Contr.*, vol. 70, no. 4, pp. 2722–2729, Apr. 2025, doi: 10.1109/TAC.2024.3496572.
[4] W. H. Organization, "Improving the Quality of Health Services - Tools and Resources," *WHO Serv. Deliv. Saf. Dep.*, pp. 1–59, 2018.
[5] T. Purchase, P. Bowie, P. Hibbert, R. G. Krishnan, and A. Carson-Stevens, "Human Factors to Improve Patient Safety," *Patient Saf. A Case-based Innov. Playb. Safer Care Second Ed.*, pp. 45–60, Jan. 2023, doi: 10.1007/978-3-031-35933-0_4.
[6] S. Thukral *et al.*, "Diagnosis of Safety Problems Using Safety Analyst for Efficient and Effective Safety Management," 2013.
[7] M. Mohamad, J. P. Steghöfer, E. Knauss, and R. Scandariato, "Managing security evidence in safety-critical organizations," *J. Syst. Softw.*, vol. 214, p. 112082, Aug. 2024, doi: 10.1016/J.JSS.2024.112082.
[8] R. Sadeghi and F. Goerlandt, "A proposed validation framework for the system theoretic process analysis (STPA) technique," *Saf. Sci.*, vol. 162, p. 106080, Jun. 2023, doi: 10.1016/J.SSCI.2023.106080.
[9] X. Chen *et al.*, "Empowering Domain Experts With Formal Methods for Consistency Verification of Safety Requirements," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15146–15157, Dec. 2023, doi: 10.1109/TITS.2023.3324022.
[10] A. Ait Wakrime and Y. Ouhammou, "Advances in modeling, verification and testing of safety-critical software architectures," *Innov. Syst. Softw. Eng.*, vol. 18, no. 4, pp. 483–484, Dec. 2022, doi: 10.1007/S11334-022-00493-Z/METRICS.
[11] A. Cimatti, M. Roveri, A. Susi, and S. Tonetta, "Formalization and Validation of Safety-Critical Requirements," *Electron. Proc. Theor. Comput. Sci. EPTCS*, vol. 20, pp. 68–75, Jun. 2012, doi: 10.4204/EPTCS.20.7.
[12] J. Fox *et al.*, "Expert systems for safety-critical applications: theory, technology and applications," in *IEE Colloquium on Knowledge-Based Systems for Safety Critical Applications*, 1994, pp. 5/1-5/5.
[13] T. Segreto, "Knowledge-Based System," *CIRP Encycl. Prod. Eng.*, pp. 997–1001, 2019, doi: 10.1007/978-3-662-53120-4_6557.
[14] J. P. Steghöfer, E. Knauss, J. Horkoff, and R. Wohlrab, "Challenges of Scaled Agile for Safety-Critical Systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11915 LNCS, pp. 350–366, 2019, doi: 10.1007/978-3-030-35333-9_26.
[15] J. Fox and D. Robertson, "Industrial use of safety related expert systems," *Heal. Saf. Exec.*, vol. 296, 2000.
[16] S. G. Tzafestas, A. I. Kokkinaki, and K. P. Valavanis, "An Overview of Expert Systems," *Expert Syst. Eng. Appl.*, pp. 3–24, 1993, doi: 10.1007/978-3-642-84048-7_1.
[17] L. E. G. Martins and T. Gorschek, "Requirements Engineering for Safety-Critical Systems: An Interview Study with Industry Practitioners," *IEEE Trans. Softw. Eng.*, vol. 46, no. 04, pp. 346–361, 2020, doi: 10.1109/TSE.2018.2854716.
[18] J. Wu, X. Zhang, M. Song, and M. Lind, "Challenges in Functional Modelling for Safety and Risk Analysis," in *Proceeding of the 33rd European Safety and Reliability Conference*, 2023, pp. 1892–1899.
[19] B. Hendrix, T. E. Lewis, M. Emery, and B. Rachele, "Model Based Functional Safety – How Functional Is It?," *J. Syst. Saf.*, vol. 57, no. 2, pp. 32–38, Jun. 2022, doi: 10.56094/JSS.V57I2.192.
[20] A. Nouri, B. Cabrero-Daniel, F. Torner, H. Sivencrona, and C. Berger, "Engineering Safety Requirements for Autonomous Driving with Large Language Models," *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 218–228, 2024, doi: 10.1109/RE59067.2024.00029.
[21] J. G. Sheehan, J. L. Howe, A. Fong, S. A. Krevat, and R. M. Ratwani, "Usability and Accessibility of Publicly Available Patient Safety Databases," *J. Patient Saf.*, vol. 18, no. 6, pp. 565–569, Sep. 2022, doi: 10.1097/PTS.0000000000001018.
[22] S. Gupta *et al.*, "Industrial Expectations of a Pure Component Database for Thermodynamic and Transport Properties," *Ind. Eng. Chem. Res.*, vol. 61, no. 42, pp. 15514–15553, Oct. 2022, doi: 10.1021/acs.iecr.2c01642.
[23] M. B. McKinnon and G. T. Bellamy, "Fire Safety Research Institute Materials and Products database—A resource to support fire modeling," *J. Fire Sci.*, vol. 42, no. 3, pp. 175–216, May 2024, doi: 10.1177/07349041241235566/ASSET/9B5829C4-6322-403A-90CD-3A5D5335B63E/ASSETS/IMAGES/LARGE/10.1177_07349041241235566-FIG16.JPG.
[24] A. Maurya and D. Kumar, "Reliability of safety-critical systems: A state-of-the-art review," *Qual. Reliab. Eng. Int.*, vol. 36, no. 7, pp. 2547–2568, Nov. 2020, doi: 10.1002/QRE.2715.
[25] V. Nguyen Tran, L. Vu Tran, V. Nguyen Tran, and D. Ngoc Vu, "Hazard Analysis Methods for Software Safety Requirements

Engineering," *ACM Int. Conf. Proceeding Ser.*, pp. 11–18, Jan. 2022, doi: 10.1145/3520084.3520087;PAGE:STRING:ARTICLE/CHAPTER.

[26] D. C. Jensen and I. Y. Tumer, "Modeling and Analysis of Safety in Early Design," *Procedia Comput. Sci.*, vol. 16, pp. 824–833, Jan. 2013, doi: 10.1016/J.PROCS.2013.01.086.

[27] K. Hamidi, O. Malasse Dr., and J. F. Aubry Prof., "Contribution to an improvement of quantitative evaluation model for reliability of safetv-related functions," *IEEE Int. Symp. Ind. Electron.*, vol. 1, pp. 115–120, 2004, doi: 10.1109/ISIE.2004.1571792.

[28] A. Abdullah, R. A. Bakar, K. Gunaratnam, F. Hujainah, and M. F. Abdul Farid, "Safety Property Attributes in Critical Systems for Requirement Specification: A Review," *8th Int. Conf. Softw. Eng. Comput. Syst. ICSECS 2023*, pp. 481–486, 2023, doi: 10.1109/ICSECS58457.2023.10256294.

[29] F. Barez, H. Hasanbieg, and A. Abbate, "System III: Learning with Domain Knowledge for Safety Constraints," no. NeurIPS, pp. 1–10, Apr. 2023, Accessed: Aug. 28, 2025. [Online]. Available: https://arxiv.org/pdf/2304.11593.

[30] Y. Liu, J. M. Wu, M. Avdeev, and S. Q. Shi, "Multi-Layer Feature Selection Incorporating Weighted Score-Based Expert Knowledge toward Modeling Materials with Targeted Properties," *Adv. Theory Simulations*, vol. 3, no. 2, p. 1900215, Feb. 2020, doi: 10.1002/ADTS.201900215;SUBPAGE:STRING:FULL.

[31] I. Babeshko, O. Illiashenko, V. Kharchenko, and K. Leontiev, "Towards Trustworthy Safety Assessment by Providing Expert and Tool-Based XMECA Techniques," *Math. 2022, Vol. 10, Page 2297*, vol. 10, no. 13, p. 2297, Jun. 2022, doi: 10.3390/MATH10132297.

[32] "UMN Critical Systems Group (CriSys)." https://crisys.cs.umn.edu/gpca.shtml (accessed Aug. 28, 2025).

[33] "The Generic Infusion Pump (GIP)." https://rtg.cis.upenn.edu/gip/#Publications (accessed Aug. 28, 2025).