Technical Letter Report

TLR-RES/DE/REB-2022-13

Autonomous Researcher Feasibility Studies

Date:

December 16, 2022

Jesse Carlson, Matthew Homiack, Raj Iyengar

Reactor Engineering Branch
Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555–0001

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed in this publication, or represents that its use by such third party complies with applicable law.

This report does not contain or imply legally binding requirements. Nor does this report establish or modify any regulatory guidance or positions of the U.S. Nuclear Regulatory Commission and is not binding on the Commission.

EXECUTIVE SUMMARY

Advances in technologies such as artificial intelligence (AI), machine learning (ML), and data analytics have the potential to enable the creation of autonomous researchers (i.e., bots) that can assist the U.S. Nuclear Regulatory Commission (NRC) staff. This report is the outcome of an exploration of AI/ML applications, supported by the future-focused research project on “RESbot – A Web-Based Bot to Aid RES Researchers.” This project was one of six research proposals selected in Fiscal Year 2021 as part of the NRC’s Future-Focused Research Program, which supports the NRC’s vision of becoming a modern, risk-informed regulator by exploring “over-the-horizon” issues and anticipating future regulatory needs. The report summarizes feasibility studies to develop research bots for two applications: (1) knowledge mining, and (2) intelligent search of the numerical solution space for modeling and simulation.

For the knowledge mining application, the NRC staff explored the viability of using natural language processing (NLP) capabilities available in large commercial platforms to search through a collection of documents in the molten salt reactor domain and answer technical questions. The commercial tools showed promise for this type of application; however, the results from this study were inconclusive. Nonetheless, the staff observed that training the NLP models requires a significant initial investment of human resources, although this could be reduced somewhat by computer-guided processes and an effective user interface and user experience. There is accelerated progress in developing AI/ML solutions for NLP in the commercial sector, such that it is likely that the type of bot envisioned for this project may be fully functional within 5 years. As such, other commercial models may be worth exploring in future work.

For the intelligent search application, the NRC staff developed a bot to interface with one of its safety codes—the Extremely Low Probability of Rupture probabilistic fracture mechanics code. This interface allowed the NRC staff to couple the Extremely Low Probability of Rupture code with open-source machine learning models. The machine learning results were then used to understand the importance of the input variables and to automate sensitivity analyses and sensitivity studies. Such applications have immediate use to support research efforts and to review licensing applications that rely on probabilistic fracture mechanics consistent with the guidance in NRC Regulatory Guide 1.245, “Preparing Probabilistic Fracture Mechanics Submittals,” Revision 0, issued January 2002. More promise was shown with this investigation, and the approach could also be extended to other NRC probabilistic modeling and simulation applications.

Through these two sample applications, the NRC staff explored aspects of AI, specifically NLP and ML, to automate various research tasks. Although NLP applications would require more exploration, both aspects show promise for assisting the NRC staff in more efficiently and effectively fulfilling the agency’s mission. This report is for internal NRC use, as it is an exploratory study and primarily intended to build knowledge and capabilities in AI/ML technology applications.

ACKNOWLEDGEMENTS

Many individuals contributed to the success of this project to explore the development of autonomous researchers to assist the NRC staff. For the knowledge mining use case investigations, the authors specifically extend thanks to the staff from the Office of the Chief Information Officer, particularly Bob Randall and his team, and staff members of the two vendors who facilitated investigations using commercial cloud services. Dr. Wendy Reed from the NRC staff is also acknowledged for contributing her expertise on molten salt reactors. For the intelligent search use case investigation, the authors thank Messrs. Stephen Verzi and Joseph Lubars from Sandia National Laboratories, under contact to the NRC, for developing the intelligent search algorithm using open-source software. The authors additionally thank Dr. Cédric Sallaberry of Engineering Mechanics Corporation of Columbus, also under contract to the NRC, for his expert review of the intelligent search algorithm results.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ADAMS	Agencywide Document Documents Access and Management System
AI	artificial intelligence
BEAR	Bayesian experimental autonomous researcher
CI	confidence interval
DOE	U.S. Department of Energy
FFR	future-focused research
SAR	safety analysis report
GDC	general design criterion
ID	identification
JSON	JavaScript Object Notation
ML	machine learning
MSR	molten salt reactor
NER	named entity recognition
NLP	natural language processing
NRC	U.S. Nuclear Regulatory Commission
PFM	probabilistic fracture mechanics
PWSCC	primary water stress corrosion cracking
QoI	quantity of interest
RES	Office of Nuclear Regulatory Research
RG	regulatory guide
SME	subject matter expert
SNL	Sandia National Laboratories
TLR	technical letter report
WRS	welding residual stresses
xLPR	Extremely Low Probability of Rupture

1 INTRODUCTION

1.1 Background on Autonomous Researchers

In recent years, tremendous advances in digital technologies have enabled exponentially more data to be generated and stored. This has created new problems for those who perform research and need to use the most up to date information, techniques, and methods; data and documents are being generated faster than they can be humanly analyzed. Staff from the U.S. Nuclear Regulatory Commission (NRC) are no exception and review enormous amounts of documents in their daily work, from regulations to technical journal articles to safety analysis reports (SARs).

In 2004, King et al. [1] explored the possibility of automating a scientific investigation through physical implementation of a robotic system that applies techniques from artificial intelligence (AI). Based on this earlier work, King et al. [2], along with other scientists, created a robot scientist, called Adam, who was able to perform independent experiments to test hypotheses and analyze the results. Since that time, there have been some strides made in the development of "robot scientists". For example, Williams et al. [3] designed a robot scientist 'Eve' to make drug discovery techniques to develop data and advanced scientific knowledge.

More recently, Burger et al. [4] developed a robot chemist to assist in experimental searches even with challenging sets of parameters, such as sample types, instruments, and measurements. Using a Bayesian search algorithm, they were able to succeed in making the robot operate autonomously over 8 days, performing 688 experiments, within a ten-variable experimental space. Gongora et al. [5] developed, what they termed as, Bayesian experimental autonomous researcher (BEAR), which uses a Bayesian scheme for topological optimization and high-throughput automated experimentation, to address the vastness of additive manufacturing design space and automatically decide the experiments to perform. They show that BEAR not only performs experiments rapidly, but it also leverages iterative experimentation by selecting experiments based on all available results. Their results demonstrate the utility and value of machine learning (ML) in experimental fields where data are sparse.

1.2 Future Focused Research Proposal

The Office of Nuclear Regulatory Research (RES) future-focused research (FFR) program supports the NRC's vision of becoming a modern, risk-informed regulator by exploring "over-the-horizon" issues and anticipating future regulatory needs. The FFR program facilitates the identification, prioritization, performance, and monitoring of research activities intended to help the NRC prepare for upcoming challenges.

Motivated by the previous work on automated researchers, the NRC's Office of Nuclear Regulatory Research (RES) carried out an FFR project aimed at assessing the feasibility of developing its own autonomous researchers, or "RESbots," to assist the NRC staff in conducting research-related activities. The proposed project suggested developing autonomous researchers for the following use cases:

1) Compile useful data from a variety of sources, including databases and communities of practice, to aid NRC staff assessment of significant and emergent topics. Examples include reactor pressure vessel embrittlement issues, materials degradation issue for long-term operations, risk assessments, and preparation for advanced reactor licensing.
2) Perform modeling and simulation to incorporate data and model uncertainties using an appropriately modified autonomous researcher approach adopted in Gongora et al. [5]. It is noted that use of Bayesian search algorithms to zone in on the regions of interest in a vast solution space is not a novel concept to NRC staff.
3) Accelerate the review process for NRC technical letter reports (TLRs). Review processes often rely on staff compilation and disposition of review comments from many staff members from RES and customer offices. Often, two sets of reviews are conducted. From initial draft to final issuance of a TLR can range from 4 to 12 months. While some of this time is essential, much of it is spent on collection, compilation, and transmittal of comments. An autonomous researcher could significantly speed up this review process, not only to supplement the manual process of comment compilation, but also to intelligently analyze the comments and offer advice for eventual resolution.

1.3 Selected Use Cases

After performing some initial scoping studies, RES staff decided to pursue only the first two proposed use cases. Whereas Use Case 1 focuses on language understanding, Use Case 2 focuses on the understanding of numerical data. Due to available resources and complexity of the problem, Use Case 3 was not pursued.

Use Case 1 involved mining a repository of publicly available documents for detailed technical information as a proof-of-concept. A repository of 100 documents was gathered about molten salt reactors (MSRs) from a variety of public sources, including academic journals, Department of Energy (DOE) databases, NRC-sponsored reports, and NRC-authored documents. These documents cover a wide range of MSR subtopics, such as corrosion, chemistry, purification, and waste forms. After the repository was assembled, RES staff subject matter experts (SMEs) assembled a list of technical questions that a future researcher would be likely to ask about MSRs. It was assumed that the questions would be nuanced by some level of knowledge and understanding of the technical domain (i.e., the questions would be more sophisticated than a simple search for the presence of a specific word or phrase). The goal of Use Case 1 was for the RESbot to return reasonable and technically accurate answers to these questions.

Use Case 2 involved developing an autonomous researcher for intelligent search of the problem space to support modeling and simulation applications. Whereas Use Case 1 focused on language understanding, Use Case 2 focused on data understanding. The staff selected the Extremely Low Probability of Rupture (xLPR) probabilistic fracture mechanics (PFM) code as the modeling and simulation environment for the Use Case 2 investigations. New software modules were written to interface with the xLPR code and employ ML tools to analyze its input and output data and generate insights to better understand the solution space, particularly the most influential inputs. Use Case 2 thus represented the marriage of PFM and ML technologies.

This TLR summarizes the RES staff’s explorations of the two use cases. The report is for internal NRC use, as it is an exploratory study and primarily intended to build knowledge and capabilities in AI/ML technology applications.

2 USE CASE 1 INVESTIGATION—KNOWLEDGE MINING

2.1 Background

The objective of Use Case 1 of the RESbot project was to develop an autonomous researcher to mine a repository of publicly available documents for detailed technical information.

2.1.1 Regulatory Context

Robotic technology has the potential to enhance efficiency and reliability for the staff to carry out the NRC's mission in many regulatory settings, from research to licensing to guidance development. The NRC staff perform research to confirm information and data provided by licensees and to be ready for imminent and future applications of current and advanced technologies impacting nuclear safety. Thus, dealing with voluminous data and operating experience for analyses, modeling and simulation, risk assessments, and technical basis development, is part and parcel of the NRC staff's daily work. Additionally, NRC staff in licensing need to be able to review regulatory and licensing documents, such as safety analysis reports (SARs), to quickly locate key technical parameters and assumptions. A major opportunity where AI/ML technology—specifically NLP and knowledge mining capabilities—can provide significant knowledge transfer benefits is in facilitating the onboarding process for new hires.

2.1.2 NLP Technology

Natural language processing (NLP) forms the fundamental technological backbone for the first RESbot use case on knowledge mining. According to Leong and Jordan [6], since language can be represented by a set of rules, computer programs can be created to organize, classify, and predict language according to those rules (learned or programmed and pre-determined). Large bodies of text can be examined using tagged entities or a dictionary of specific search terms, and NLP programs can apply the set of rules to construct a wide spectrum of desired outputs, whether it be interpretation, analysis, understanding, question answering, or text creation.

2.2 Approach

The RES staff's approach for investigating Use Case 1 was the following:

- select a technical area and identify related records
- develop potential NRC staff questions about the domain
- pursue development of RESbots using commercial tools

While it was desirable to perform the knowledge mining from a set of random documents, the RES staff realized that the state-of-technology was not mature enough. Hence, to simplify the use case, the RES staff chose to build a document repository based on the single topic of MSRs. This topic was selected because it has been an area of substantial interest, there are a wide range of applications and technical challenges, and MSRs have significant operational differences with respect to traditional light water reactor technologies. After selection of this topic, the RES staff collected 100 publicly available documents from a wide variety of sources (e.g., DOE databases, NRC's Agencywide Documents Access and Management System, and academic journals). The RES staff developed a list of questions that a new NRC staff member might have and would want to be able to answer. These steps were done in coordination with MSR SMEs who provided significant guidance, input, and direction. The primary intent of this work was to assess the capabilities and limitations of available commercial tools to perform knowledge mining and summarization of key topics from a myriad of documents.

Development of one RESbot was pursued using the cloud service of a large commercial vendor (Vendor 1) that provides indexing and querying capabilities enabled by built-in AI algorithms and uses advanced ML techniques to understand user intent and contextually rank the most relevant search results.

Development of a second RESbot was pursued using the cloud service of a large commercial vendor (Vendor 2) that provides a question-answering computer system capable of answering questions posed in natural language. Its deep conversational AI uses a chatbot that uses natural language understanding to interact with users through common dialogue. A search feature uses advanced AI to scan structured and unstructured data and extract relevant insights, and another feature connects to data sources.

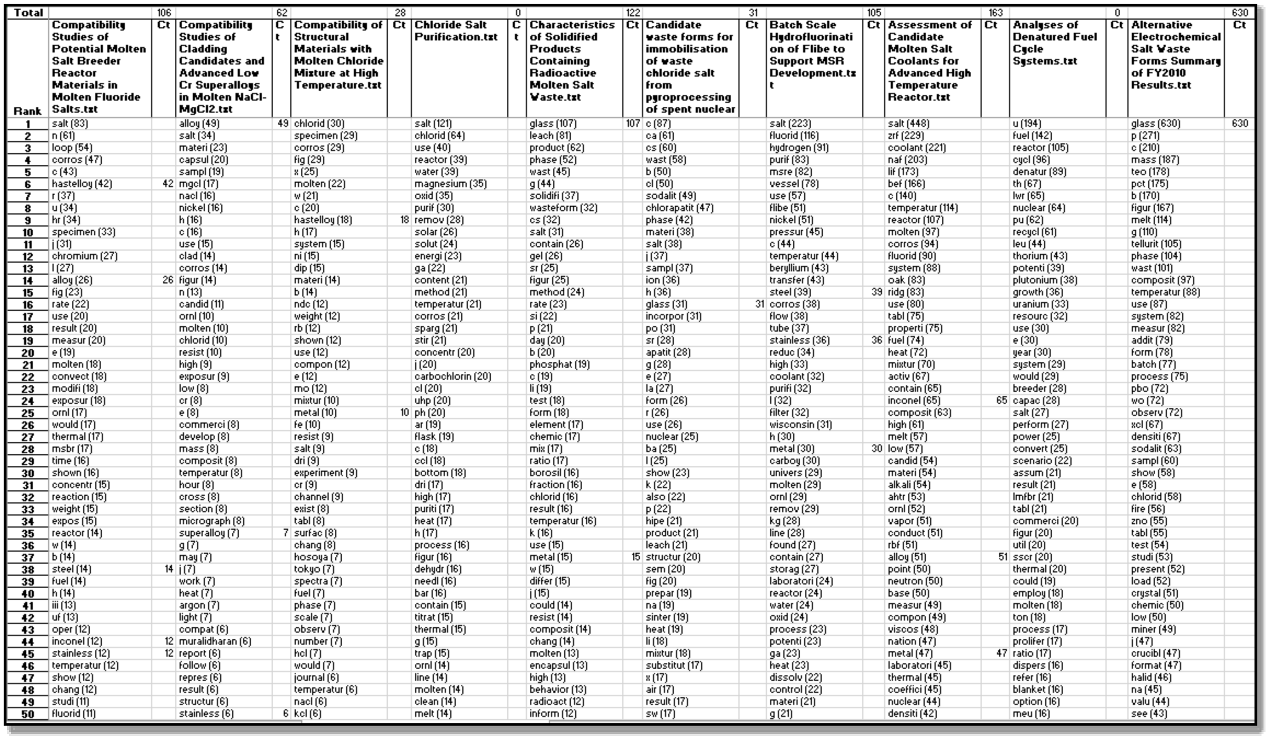
2.3 Results and Discussion

2.3.1 Vendor 1 RESbot

The RES staff met with representatives from Vendor 1 to discuss its objectives for the RESbot project. The vendor offered a limited term subscription to use its cloud service, which the RES staff used to pursue development of a RESbot for Use Case 1. Development of the RESbot in this environment was primarily the responsibility of the RES staff; however, an SME from Vendor 1 provided some limited guidance to help navigate the various systems and settings to conduct the testing.

As a first step, the RES staff uploaded the 100 MSR documents into Vendor 1's cloud storage in the file extensions in which they had had been originally retrieved (e.g., *.pdf or *.docx). The system was capable of handling and reading the documents in any of the common file types. At this point, the RES staff chose to explore the use of a recently developed tool for training and customizing programs to the unique terms, phrases, and vernacular of a particular domain. This tool uses custom named entity recognition (NER) and allows users to build custom AI models to extract domain-specific entities by iteratively labeling data, training, evaluating, and improving model performance before making it available for consumption. Prior to labeling the data, the documents needed to be in a text format to be compatible with the NER tools. The RES staff manually converted the *.pdf documents to text documents using the native Adobe converter tool. It is noted that Vendor 1's cloud service includes a feature to convert *.pdf documents to plain text files; however, this feature was not included in the subscription. As many of the documents were from scanned images, the RES staff opted to correct certain common errors that resulted from the conversion process. These corrections were made through development of a simple Python routine that scanned for and replaced the artifacts. The RES staff then decided on the following four custom named entities associated with MSRs: (1) Reactor Name, (2) Fuel, (3) Salt, and (4) Component Material.

Considering the large number of documents, the RES staff was faced with the immediate question of identifying which would be the best candidates for use in tagging the custom named entities. That is, it would have been inefficient to read through an entire document only to find that it contained no examples of the selected named entities. To help answer this question, the staff wrote a routine in Python that leverages the Natural Language Toolkit package. Functions from the Python Standard Library were used to convert the text in each document to lowercase and remove punctuation. Then, functions from the Natural Language Toolkit package were used to tokenize the text and remove common English stopwords, such as “and,” “are”, “is,” and “the.” The Porter Stemmer algorithm was then used to remove morphological affixes, thereby only leaving the word stems. For example, the stem of “purification” is “purif.” Finally, the top 50 stems by frequency were listed for each document in a spreadsheet. Figure 2-1 shows a selection of the results.

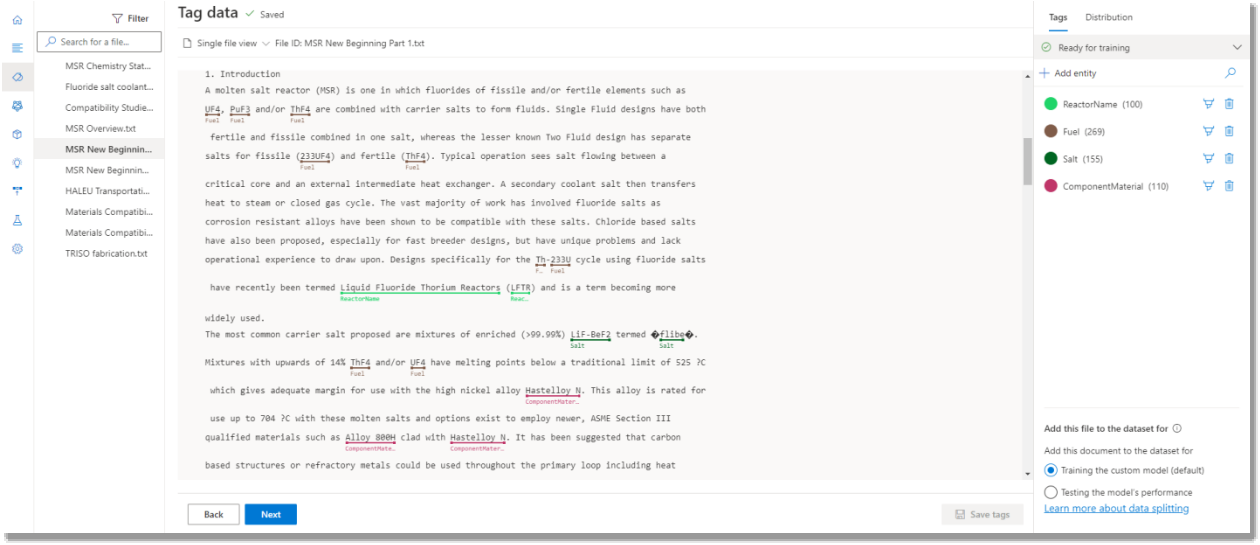


Total		106		62		28		0		122		31		105		163		0		630
Rank	Compatibility Studies of Potential Molten Salt Breeder Reactor Materials in Molten Fluoride Salts.txt	Ct	Compatibility Studies of Cladding Candidates and Advanced Low Cr Superalloys in Molten NaCl-MgCl2.txt	Ct	Compatibility of Structural Materials with Molten Chloride Mixture at High Temperature.txt	Ct	Chloride Salt Purification.txt	Ct	Characteristics of Solidified Products Containing Radioactive Molten Salt Waste.txt	Ct	Candidate waste forms for immobilisation of waste chloride salt from pyroprocessing of spent nuclear	Ct	Batch Scale Hydrofluorination of Flibe to Support MSR Development.txt	Ct	Assessment of Candidate Molten Salt Coolants for Advanced High Temperature Reactor.txt	Ct	Analyses of Denatured Fuel Cycle Systems.txt	Ct	Alternative Electrochemical Salt Waste Forms Summary of FY2010 Results.txt	Ct
1	salt (83)		alloy (49)	49	chlorid (30)		salt (121)		glass (107)	107	c (87)		salt (223)		salt (448)		u (194)		glass (630)	630
2	n (61)		salt (34)		specimen (29)		chlorid (64)		leach (81)		ca (61)		fluorid (116)		zrf (229)		fuel (142)		p (271)	
3	loop (54)		materi (23)		corros (29)		use (40)		product (62)		cs (60)		hydrogen (91)		coolant (221)		reactor (105)		c (210)	
4	corros (47)		capsul (20)		fig (29)		reactor (39)		phase (52)		wast (58)		purif (83)		naf (203)		cycl (96)		mass (187)	
5	c (43)		sampl (19)		x (25)		water (39)		wast (45)		b (50)		msre (82)		lif (173)		denatur (89)		teo (178)	
6	hastelloy (42)	42	mgcl (17)		molten (22)		magnesium (35)		g (44)		cl (50)		vessel (78)		bef (166)		th (67)		pct (175)	
7	r (37)		nacl (16)		w (21)		oxid (35)		solidifi (37)		sodalit (49)		use (57)		c (140)		lwr (65)		b (170)	
8	u (34)		nickel (16)		c (20)		purif (30)		wasteform (32)		chlorapatit (47)		flibe (51)		temperatur (114)		nuclear (64)		figur (167)	
9	hr (34)		h (16)		hastelloy (18)	18	remov (28)		cs (32)		phase (42)		nickel (51)		reactor (107)		pu (62)		melt (114)	
10	specimen (33)		c (16)		h (17)		solar (26)		salt (31)		materi (38)		pressur (45)		molten (97)		recycl (61)		g (110)	
11	j (31)		use (15)		system (15)		solut (24)		contain (26)		salt (38)		c (44)		corros (94)		leu (44)		tellurit (105)	
12	chromium (27)		clad (14)		ni (15)		energi (23)		gel (26)		j (37)		temperatur (44)		fluorid (90)		thorium (43)		phase (104)	
13	l (27)		corros (14)		dip (15)		ga (22)		sr (25)		sampl (37)		beryllium (43)		system (88)		potenti (39)		wast (101)	
14	alloy (26)	26	figur (14)		materi (14)		content (21)		figur (25)		ion (36)		transfer (43)		oak (83)		plutonium (38)		composit (97)	
15	fig (23)		n (13)		b (14)		method (21)		method (24)		h (36)		steel (39)	39	ridg (83)		growth (36)		temperatur (88)	
16	rate (22)		candid (11)		ndc (12)		temperatur (21)		rate (23)		glass (31)	31	corros (38)		use (80)		uranium (33)		use (87)	
17	use (20)		ornl (10)		weight (12)		corros (21)		si (22)		incorpor (31)		flow (38)		tabl (75)		resourc (32)		system (82)	
18	result (20)		molten (10)		rb (12)		sparg (21)		p (21)		po (31)		tube (37)		properti (75)		use (30)		measur (82)	
19	measur (20)		chlorid (10)		shown (12)		stir (21)		day (20)		sr (28)		stainless (36)	36	fuel (74)		e (30)		addit (79)	
20	e (19)		resist (10)		use (12)		concentr (20)		b (20)		apatit (28)		reduc (34)		heat (72)		year (30)		form (78)	
21	molten (18)		high (9)		compon (12)		j (20)		phosphat (19)		g (28)		high (33)		mixtur (70)		system (29)		batch (77)	
22	convect (18)		exposur (9)		e (12)		carbochlorin (20)		c (19)		e (27)		coolant (32)		activ (67)		would (29)		process (75)	
23	modifi (18)		low (8)		mo (12)		cl (20)		li (19)		la (27)		purifi (32)		contain (65)		breeder (28)		pbo (72)	
24	exposur (18)		cr (8)		mixtur (10)		uhp (20)		test (18)		form (26)		l (32)		inconel (65)	65	capac (28)		wo (72)	
25	ornl (17)		e (8)		metal (10)	10	ph (20)		form (18)		r (26)		filter (32)		composit (63)		salt (27)		observ (72)	
26	would (17)		commerci (8)		fe (10)		ar (19)		element (17)		use (26)		wisconsin (31)		high (61)		perform (27)		xcl (67)	
27	thermal (17)		develop (8)		resist (9)		flask (19)		chemic (17)		nuclear (25)		h (30)		melt (57)		power (25)		densiti (67)	
28	msbr (17)		mass (8)		salt (9)		c (18)		mix (17)		ba (25)		metal (30)	30	low (57)		convert (25)		sodalit (63)	
29	time (16)		composit (8)		dri (9)		ccl (18)		ratio (17)		l (25)		carboy (30)		candid (54)		scenario (22)		sampl (60)	
30	shown (16)		temperatur (8)		experiment (9)		bottom (18)		borosil (16)		show (23)		univers (29)		materi (54)		assum (21)		show (58)	
31	concentr (15)		hour (8)		cr (9)		dri (17)		fraction (16)		k (22)		molten (29)		alkali (54)		result (21)		e (58)	
32	reaction (15)		cross (8)		channel (9)		high (17)		chlorid (16)		also (22)		ornl (29)		ahtr (53)		lmfbr (21)		chlorid (58)	
33	weight (15)		section (8)		exist (8)		puriti (17)		result (16)		p (22)		remov (29)		ornl (52)		tabl (21)		fire (56)	
34	expos (15)		micrograph (8)		tabl (8)		heat (17)		temperatur (16)		hipe (21)		kg (28)		vapor (51)		commerci (20)		zno (55)	
35	reactor (14)		superalloy (7)	7	surfac (8)		h (17)		k (16)		product (21)		line (28)		conduct (51)		figur (20)		tabl (55)	
36	w (14)		g (7)		chang (8)		process (16)		use (15)		leach (21)		found (27)		rbf (51)		util (20)		test (54)	
37	b (14)		may (7)		hosoya (7)		figur (16)		metal (15)	15	structur (20)		contain (27)		alloy (51)	51	sscr (20)		studi (53)	
38	steel (14)	14	j (7)		tokyo (7)		dehydr (16)		w (15)		sem (20)		storag (27)		point (50)		thermal (20)		present (52)	
39	fuel (14)		work (7)		spectra (7)		needl (16)		differ (15)		fig (20)		laboratori (24)		neutron (50)		could (19)		load (52)	
40	h (14)		heat (7)		fuel (7)		bar (16)		j (15)		prepar (19)		reactor (24)		base (50)		employ (18)		crystal (51)	
41	iii (13)		argon (7)		phase (7)		contain (15)		could (14)		na (19)		water (24)		measur (49)		molten (18)		chemic (50)	
42	uf (13)		light (7)		scale (7)		titrat (15)		resist (14)		sinter (19)		oxid (24)		compon (49)		ton (18)		low (50)	
43	oper (12)		compat (6)		observ (7)		thermal (15)		composit (14)		heat (19)		process (23)		viscos (48)		process (17)		miner (49)	
44	inconel (12)	12	muralidharan (6)		number (7)		g (15)		chang (14)		li (18)		potenti (23)		nation (47)		prolifer (17)		j (47)	
45	stainless (12)	12	report (6)		hcl (7)		trap (15)		molten (13)		mixtur (18)		ga (23)		metal (47)	47	ratio (17)		crucibl (47)	
46	temperatur (12)		follow (6)		would (7)		ornl (14)		encapsul (13)		substitut (17)		heat (23)		laboratori (45)		dispers (16)		format (47)	
47	show (12)		repres (6)		journal (6)		line (14)		high (13)		x (17)		dissolv (22)		thermal (45)		refer (16)		halid (46)	
48	chang (12)		result (6)		temperatur (6)		molten (14)		behavior (13)		air (17)		control (22)		coeffici (45)		blanket (16)		na (45)	
49	studi (11)		structur (6)		nacl (6)		clean (14)		radioact (12)		result (17)		materi (21)		nuclear (44)		option (16)		valu (44)	
50	fluorid (11)		stainless (6)	6	kcl (6)		melt (14)		inform (12)		sw (17)		g (21)		densiti (42)		meu (16)		see (43)	

Figure 2-1 Top 50 most frequent word stems in a selection of documents in the MSR domain

From a quick analysis of this data, the team was able to identify that the document, “Compatibility Studies of Potential Molten Salt Breeder Reactor Materials in Molten Fluoride Salts,” with 106 occurrences of the stems “hastelloy,” “alloy,” “steel,” inconel,” and “stainless,” was a good candidate in which a variety of example materials could be tagged to help answer the question, “Which alloys are used?” Although this was a simple NLP application, the results proved useful in quickly assessing the basic content of many documents. It’s noted, however, that the utility of such an approach would be expected to decrease if the number of documents were in the thousands. Notwithstanding, this exploration revealed that Vendor 1’s approach to tagging named entities would benefit from a more guided process.

With the best candidate documents identified, the RES staff then manually carried out the process of tagging each instance of the custom named entities. In total, eight documents were tagged to ensure enough data for training. The tagging process entailed highlighting text in the documents and linking it to the appropriate custom named entity. As a result, the following quantities of tags were attained: 100 for “Reactor Name”, 269 for “Fuel”, 155 for “Salt”, and 110 for “Component Material”. An example of a tagged document in Vendor 1’s cloud service interface is shown in Figure 2-2. Shown in light green underline are the text entries tagged and associated with the named entity “Reactor Name”. They include “Liquid Fluoride Thorium Reactor” and “LFTR”. Shown in brown underline are the text entries tagged and associated with the named entity “Fuel”. They include the following chemical compounds: “UF4”, “PuF3”, “ThF4”, “233UF4”, and “Th-233U”. The entries underlined in dark green are associated with the named entity “Salt” and include “LiF-BeF2” and “flibe”. Finally, the terms underlined in magenta are linked to the named entity “Component Material”. They include “Hastelloy N” and “Alloy 800H”.



1. Introduction
A molten salt reactor (MSR) is one in which fluorides of fissile and/or fertile elements such as
UF4, PuF3 and/or ThF4 are combined with carrier salts to form fluids. Single Fluid designs have both
fertile and fissile combined in one salt, whereas the lesser known Two Fluid design has separate
salts for fissile (233UF4) and fertile (ThF4). Typical operation sees salt flowing between a
critical core and an external intermediate heat exchanger. A secondary coolant salt then transfers
heat to steam or closed gas cycle. The vast majority of work has involved fluoride salts as
corrosion resistant alloys have been shown to be compatible with these salts. Chloride based salts
have also been proposed, especially for fast breeder designs, but have unique problems and lack
operational experience to draw upon. Designs specifically for the Th-233U cycle using fluoride salts
have recently been termed Liquid Fluoride Thorium Reactors (LFTR) and is a term becoming more
widely used.
The most common carrier salt proposed are mixtures of enriched (>99.99%) LiF-BeF2 termed �flibe�.
Mixtures with upwards of 14% ThF4 and/or UF4 have melting points below a traditional limit of 525 ?C
which gives adequate margin for use with the high nickel alloy Hastelloy N. This alloy is rated for
use up to 704 ?C with these molten salts and options exist to employ newer, ASME Section III
qualified materials such as Alloy 800H clad with Hastelloy N. It has been suggested that carbon
based structures or refractory metals could be used throughout the primary loop including heat

🟢 ReactorName (100)
🟤 Fuel (269)
🟢 Salt (155)
🔴 ComponentMaterial (110)

Add this file to the dataset for
Add this document to the dataset for
● Training the custom model (default)
○ Testing the model's performance
Learn more about data splitting

Figure 2-2 Example document tagging for custom NER in Vendor 1’s cloud service

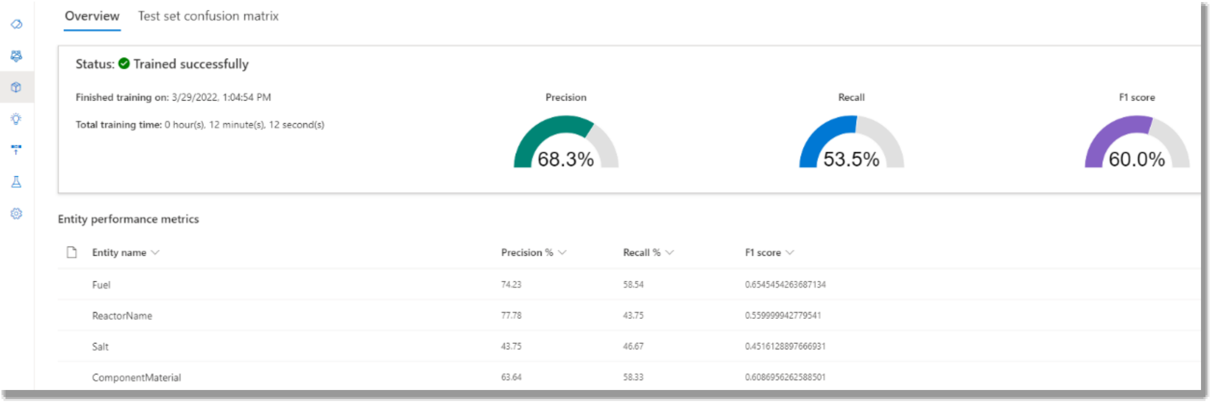
After the tagging process was complete, Vendor 1’s cloud service trained an automated ML model. The service generates three scores to indicate the efficacy of the resultant model: (1) precision, (2) recall, and (3) F1. Eq. 1, Eq. 2, and Eq. 3 below provide the definitions for the three scores, respectively.

$$Precision = \frac{No.\,True\ Positive}{No.\,True\ Positive + No.\,False\ Positive} \qquad \text{Eq. 1}$$

$$Recall = \frac{No.\,True\ Positive}{No.\,True\ Positive + No.\,False\ Negative} \qquad \text{Eq. 2}$$

$$F1\ Score = 2\frac{Precision \cdot Recall}{(Precision + Recall)} \qquad \text{Eq. 3}$$

Steps that can be taken to improve the scores include modifying the entities, further adjusting, fixing, or reviewing existing tags, or increasing the number of documents tagged. The scores after training the model on the four custom named entities and fine-tuning the tags in the eight documents were as follows (also shown in Figure 2-3): 68.3 percent for precision, 53.5 percent for recall, and 60.0 percent for the F1 score. The RES staff discussed the results with Vendor 1’s SME, who indicated that an F1 score in the range of 80 to 90 percent was typically needed for the custom NER model to enhance the search results.



Status: Trained successfully

Finished training on: 3/29/2022, 1:04:54 PM

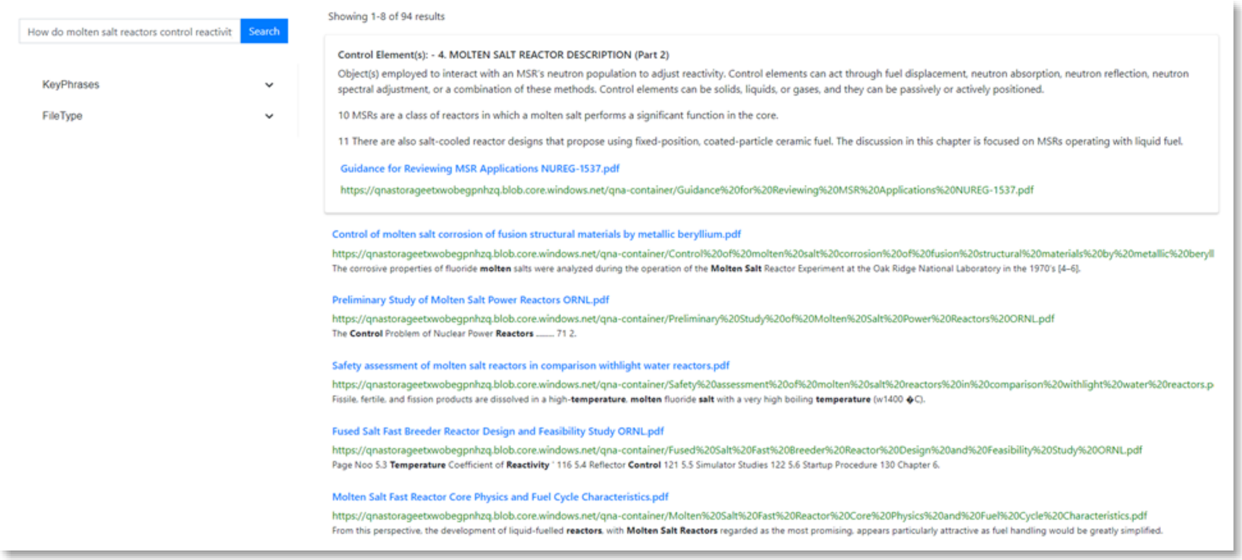
Total training time: 0 hour(s), 12 minute(s), 12 second(s)

Entity performance metrics

Entity name	Precision %	Recall %	F1 score
Fuel	74.23	58.54	0.6545454263687134
ReactorName	77.78	43.75	0.559999942779541
Salt	43.75	46.67	0.4516128897666931
ComponentMaterial	63.64	58.33	0.6086956262588501

Figure 2-3 Vendor 1 custom NER model accuracy results

The RES staff could not refine or broaden the model training due to resource constraints; therefore, it did not test the performance of Vendor 1’s search service with the custom NER model on the questions it developed. Instead, the RES staff decided to test the generic search service. Two representative responses to submitted questions are shown in Figure 2-4 and Figure 2-5. The results demonstrate the “out-of-the-box” capabilities of Vendor 1’s search service with no specific training or customization by way of incorporation of a custom NER model. This was a simple process that involved starting a new search instance and posing the predetermined questions.



Showing 1-8 of 94 results

Control Element(s): - 4. MOLTEN SALT REACTOR DESCRIPTION (Part 2)

Object(s) employed to interact with an MSR's neutron population to adjust reactivity. Control elements can act through fuel displacement, neutron absorption, neutron reflection, neutron spectral adjustment, or a combination of these methods. Control elements can be solids, liquids, or gases, and they can be passively or actively positioned.

10 MSRs are a class of reactors in which a molten salt performs a significant function in the core.

11 There are also salt-cooled reactor designs that propose using fixed-position, coated-particle ceramic fuel. The discussion in this chapter is focused on MSRs operating with liquid fuel.

Guidance for Reviewing MSR Applications NUREG-1537.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Guidance%20for%20Reviewing%20MSR%20Applications%20NUREG-1537.pdf

Control of molten salt corrosion of fusion structural materials by metallic beryllium.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Control%20of%20molten%20salt%20corrosion%20of%20fusion%20structural%20materials%20by%20metallic%20beryll

The corrosive properties of fluoride **molten** salts were analyzed during the operation of the **Molten Salt** Reactor Experiment at the Oak Ridge National Laboratory in the 1970's [4–6].

Preliminary Study of Molten Salt Power Reactors ORNL.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Preliminary%20Study%20of%20Molten%20Salt%20Power%20Reactors%20ORNL.pdf

The **Control** Problem of Nuclear Power **Reactors** .... 71 2.

Safety assessment of molten salt reactors in comparison withlight water reactors.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Safety%20assessment%20of%20molten%20salt%20reactors%20in%20comparison%20withlight%20water%20reactors.p

Fissile, fertile, and fission products are dissolved in a high-**temperature**, **molten** fluoride **salt** with a very high boiling **temperature** (w1400 �C).

Fused Salt Fast Breeder Reactor Design and Feasibility Study ORNL.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Fused%20Salt%20Fast%20Breeder%20Reactor%20Design%20and%20Feasibility%20Study%20ORNL.pdf

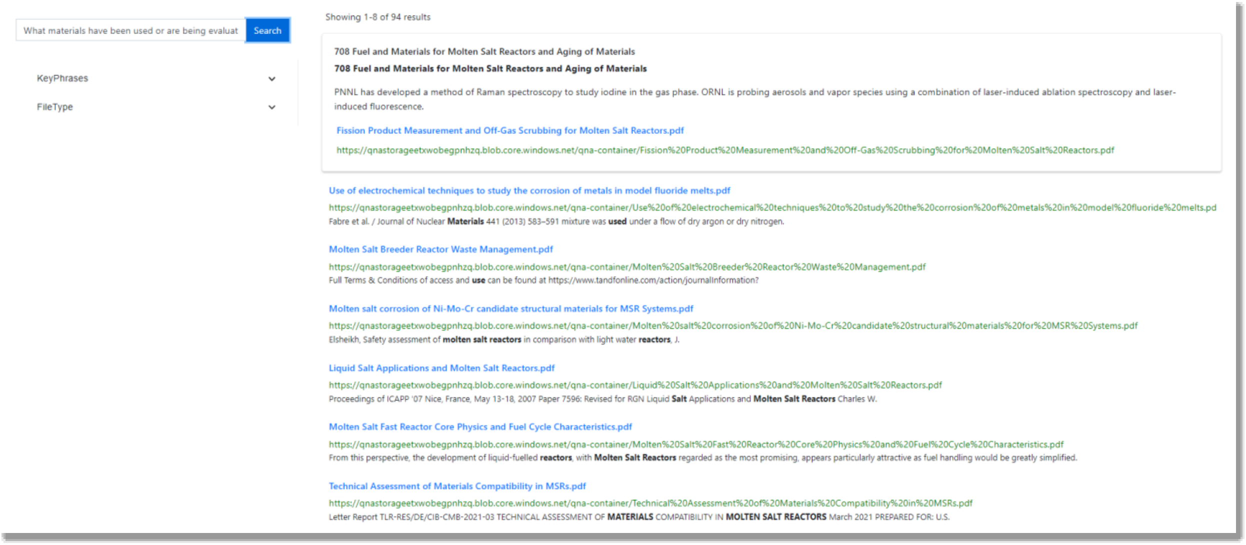
Page Noo 5.3 **Temperature** Coefficient of **Reactivity** ' 116 5.4 Reflector **Control** 121 5.5 Simulator Studies 122 5.6 Startup Procedure 130 Chapter 6.

Molten Salt Fast Reactor Core Physics and Fuel Cycle Characteristics.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Molten%20Salt%20Fast%20Reactor%20Core%20Physics%20and%20Fuel%20Cycle%20Characteristics.pdf

From this perspective, the development of liquid-fuelled **reactors**, with **Molten Salt Reactors** regarded as the most promising, appears particularly attractive as fuel handling would be greatly simplified.

Figure 2-4 Vendor 1 generic search responses to the question, "How do molten salt reactors control reactivity and temperature?"



Showing 1-8 of 94 results

708 Fuel and Materials for Molten Salt Reactors and Aging of Materials

708 Fuel and Materials for Molten Salt Reactors and Aging of Materials

PNNL has developed a method of Raman spectroscopy to study iodine in the gas phase. ORNL is probing aerosols and vapor species using a combination of laser-induced ablation spectroscopy and laser-induced fluorescence.

Fission Product Measurement and Off-Gas Scrubbing for Molten Salt Reactors.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Fission%20Product%20Measurement%20and%20Off-Gas%20Scrubbing%20for%20Molten%20Salt%20Reactors.pdf

Use of electrochemical techniques to study the corrosion of metals in model fluoride melts.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Use%20of%20electrochemical%20techniques%20to%20study%20the%20corrosion%20of%20metals%20in%20model%20fluoride%20melts.pd

Fabre et al. / Journal of Nuclear **Materials** 441 (2013) 583–591 mixture was **used** under a flow of dry argon or dry nitrogen.

Molten Salt Breeder Reactor Waste Management.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Molten%20Salt%20Breeder%20Reactor%20Waste%20Management.pdf

Full Terms & Conditions of access and **use** can be found at https://www.tandfonline.com/action/journalInformation?

Molten salt corrosion of Ni-Mo-Cr candidate structural materials for MSR Systems.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Molten%20salt%20corrosion%20of%20Ni-Mo-Cr%20candidate%20structural%20materials%20for%20MSR%20Systems.pdf

Elsheikh, Safety assessment of **molten salt reactors** in comparison with light water **reactors**, J.

Liquid Salt Applications and Molten Salt Reactors.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Liquid%20Salt%20Applications%20and%20Molten%20Salt%20Reactors.pdf

Proceedings of ICAPP '07 Nice, France, May 13-18, 2007 Paper 7596: Revised for RGN Liquid **Salt** Applications and **Molten Salt Reactors** Charles W.

Molten Salt Fast Reactor Core Physics and Fuel Cycle Characteristics.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Molten%20Salt%20Fast%20Reactor%20Core%20Physics%20and%20Fuel%20Cycle%20Characteristics.pdf

From this perspective, the development of liquid-fuelled **reactors**, with **Molten Salt Reactors** regarded as the most promising, appears particularly attractive as fuel handling would be greatly simplified.

Technical Assessment of Materials Compatibility in MSRs.pdf

https://qnastorageetxwobegpnhzq.blob.core.windows.net/qna-container/Technical%20Assessment%20of%20Materials%20Compatibility%20in%20MSRs.pdf

Letter Report TLR-RES/DE/CIB-CMB-2021-03 TECHNICAL ASSESSMENT OF **MATERIALS** COMPATIBILITY IN **MOLTEN SALT REACTORS** March 2021 PREPARED FOR: U.S.

Figure 2-5 Vendor 1 generic search responses to the question, "What materials have been used or are being evaluated for use in molten salt reactors?"

The NRC staff qualitatively evaluated the returned results. The answers to the question, “How do molten salt reactors control reactivity and temperature?” (Figure 2-4), were judged to be reasonable and appropriate. The top result clearly answers the question and discusses controlling reactivity of MSRs through interactions with the neutron population. Most of the retrieved documents are strongly related to the topic and provide helpful references for the user. The answers to the question, “What materials have been used or are being evaluated for use in molten salt reactors?” (Figure 2-5), however, were judged significantly less useful. This difference in the relevancy of the results is likely because certain terms or phrases are easier for the semantic search program to understand and find in the documents. The intent of the question was to identify structural or component materials, such as Alloy 800H, Hastelloy N, or Stainless Steel 316. The top result was not highly relevant; however, there were some relevant documents lower in the ranking. Additional resources would be needed to train and add customization to get better answers.

2.3.2 Vendor 2 RESbot

The RES staff likewise engaged representatives from Vendor 2 to discuss the objectives for the RESbot project. Motivated by the potential for broader applications of the use case, the vendor volunteered the efforts of a small team to pursue development of a concept RESbot. Pursuit of Use Case 1 with this vendor thus benefitted greatly from the expertise offered by its SMEs and provided a greater level of engagement in contrast to the resources offered by Vendor 1.

Vendor 2’s team took a top-down approach by first developing a custom user interface that would meet the use case requirements. The overarching assumption behind the approach was that a supervisor needs an intuitive tool to help onboard researchers and direct them to institutional knowledge about MSRs. The remaining assumptions are listed below in the form of tasks that said researcher would need to accomplish:

1) quickly access a large library of documents to efficiently find correct information

2) ask questions and search MSR key terms and get an accurate and succinct answer

3) see query suggestions based on the query topic(s)

4) see search results used for the answer summary they receive

5) see the document summary

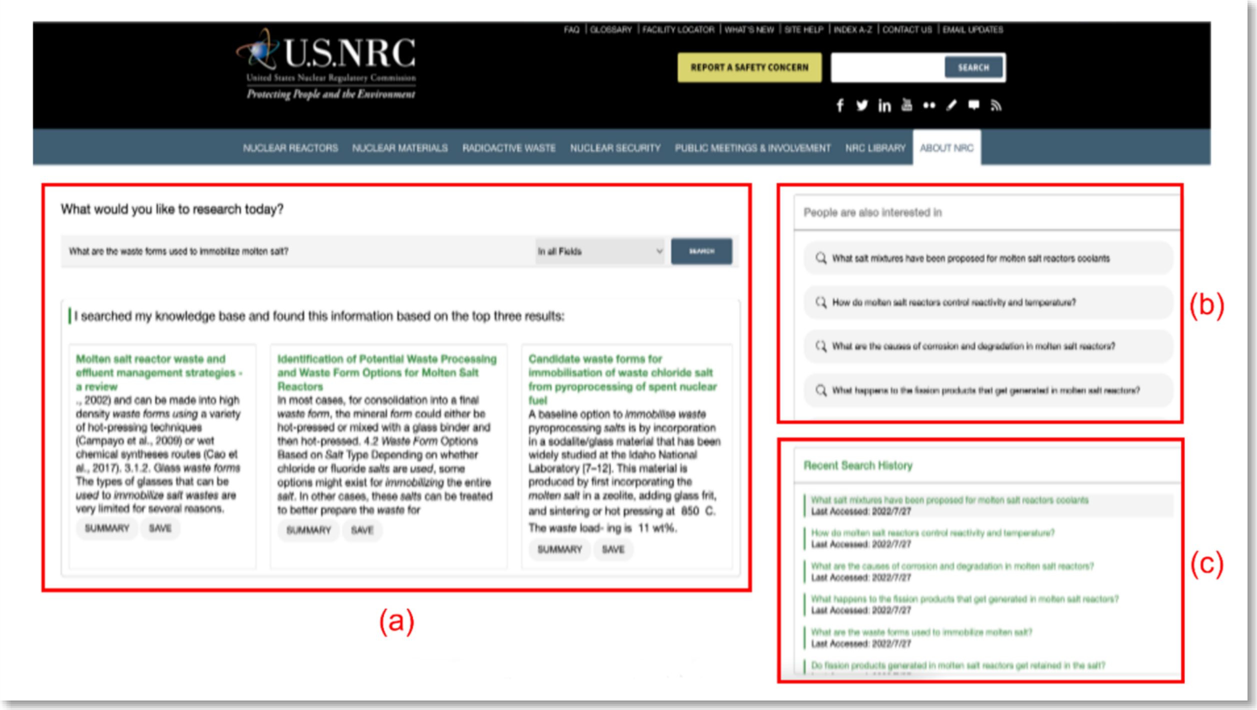
6) rate the answer accuracy to continue to train the AI and see document summaries

7) explore relevant topics and questions that others have searched

8) see search history

9) bookmark important information for future reference

Figure 2-6 shows the resultant user interface developed by Vendor 2. Inset (a) of the figure is the main feature of the interface where questions can be asked, relevant document passages retrieved, and document summaries can be requested and displayed. Inset (b) contains frequently asked questions where users can get quick answers to questions of high interest. Finally, inset (c) displays a recent search history that allows the user to access previous search results. The RES staff found this user interface to be highly functional and visually appealing.


FAQ | GLOSSARY | FACILITY LOCATOR | WHAT'S NEW | SITE HELP | INDEX A-Z | CONTACT US | EMAIL UPDATES
U.S.NRC
United States Nuclear Regulatory Commission
Protecting People and the Environment
REPORT A SAFETY CONCERN
SEARCH
NUCLEAR REACTORS
NUCLEAR MATERIALS
RADIOACTIVE WASTE
NUCLEAR SECURITY
PUBLIC MEETINGS & INVOLVEMENT
NRC LIBRARY
ABOUT NRC
In all Fields
SEARCH
SUMMARY
SAVE
(a)
(b)
(c)

What would you like to research today?

What are the waste forms used to immobilize molten salt?

I searched my knowledge base and found this information based on the top three results:

Molten salt reactor waste and effluent management strategies - a review
., 2002) and can be made into high density *waste forms using* a variety of hot-pressing techniques (Campayo et al., 2009) or wet chemical syntheses routes (Cao et al., 2017). 3.1.2. Glass *waste forms* The types of glasses that can be *used* to *immobilize salt wastes* are very limited for several reasons.

Identification of Potential Waste Processing and Waste Form Options for Molten Salt Reactors
In most cases, for consolidation into a final *waste form*, the mineral *form could* either be hot-pressed or mixed with a glass binder and then hot-pressed. 4.2 *Waste Form* Options Based on *Salt* Type Depending on whether chloride or fluoride *salts* are *used*, some options might exist for *immobilizing* the entire *salt*. In other cases, these *salts* can be treated to better prepare the *waste for*

Candidate waste forms for immobilisation of waste chloride salt from pyroprocessing of spent nuclear fuel
A baseline option to *immobilise waste* pyroprocessing *salts* is by incorporation in a sodalite/glass material that has been widely studied at the Idaho National Laboratory [7–12]. This material is produced by first incorporating the *molten salt* in a zeolite, adding glass frit, and sintering or hot pressing at 850 C. The *waste* load- ing is 11 wt%.

People are also interested in

What salt mixtures have been proposed for molten salt reactors coolants

How do molten salt reactors control reactivity and temperature?

What are the causes of corrosion and degradation in molten salt reactors?

What happens to the fission products that get generated in molten salt reactors?

Recent Search History

What salt mixtures have been proposed for molten salt reactors coolants
Last Accessed: 2022/7/27

How do molten salt reactors control reactivity and temperature?
Last Accessed: 2022/7/27

What are the causes of corrosion and degradation in molten salt reactors?
Last Accessed: 2022/7/27

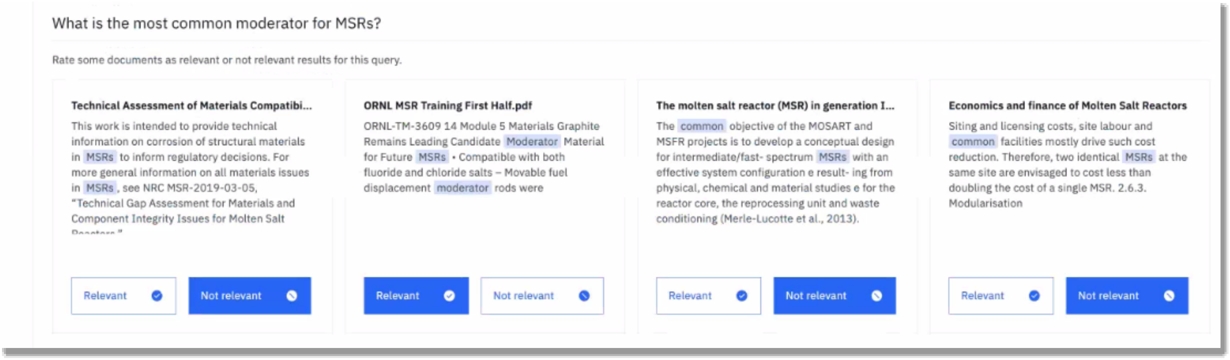
What happens to the fission products that get generated in molten salt reactors?
Last Accessed: 2022/7/27

What are the waste forms used to immobilize molten salt?
Last Accessed: 2022/7/27

Do fission products generated in molten salt reactors get retained in the salt?

Figure 2-6 RESbot user interface proposed by Vendor 2 showing (a) query feature, (b) frequency asked questions, and (c) recent search history

After uploading the collection of 100 MSR documents to Vendor 2's cloud service and providing the list of predetermined questions, the RES staff developed answers to the questions from excerpts in the documents. Vendor 2's team used the answers as a basis to conduct relevancy training for the model. To conduct relevancy training, Vendor 2's cloud service interface displays passages from the documents that a user marks as either being "relevant" or "not relevant". The results allow the cloud service to run ML models in the background to customize answers and learn what is relevant to the domain and improve the quality of answers to a given question. In this case, Vendor 2's team used the answers provided by RES staff SMEs as the basis to conduct a limited amount of relevancy training. Figure 2-7 shows a sample snapshot of relevancy training in Vendor 2's cloud service on the question, "What is the most common moderator for MSRs?" The RES staff found this relevancy training process to be a simple and efficient means of training and customizing a model to improve the quality of the search results.



What is the most common moderator for MSRs?

Rate some documents as relevant or not relevant results for this query.

Technical Assessment of Materials Compatibi...

This work is intended to provide technical information on corrosion of structural materials in MSRs to inform regulatory decisions. For more general information on all materials issues in MSRs, see NRC MSR-2019-03-05, "Technical Gap Assessment for Materials and Component Integrity Issues for Molten Salt Reactors."

ORNL MSR Training First Half.pdf

ORNL-TM-3609 14 Module 5 Materials Graphite Remains Leading Candidate Moderator Material for Future MSRs • Compatible with both fluoride and chloride salts – Movable fuel displacement moderator rods were

The molten salt reactor (MSR) in generation I...

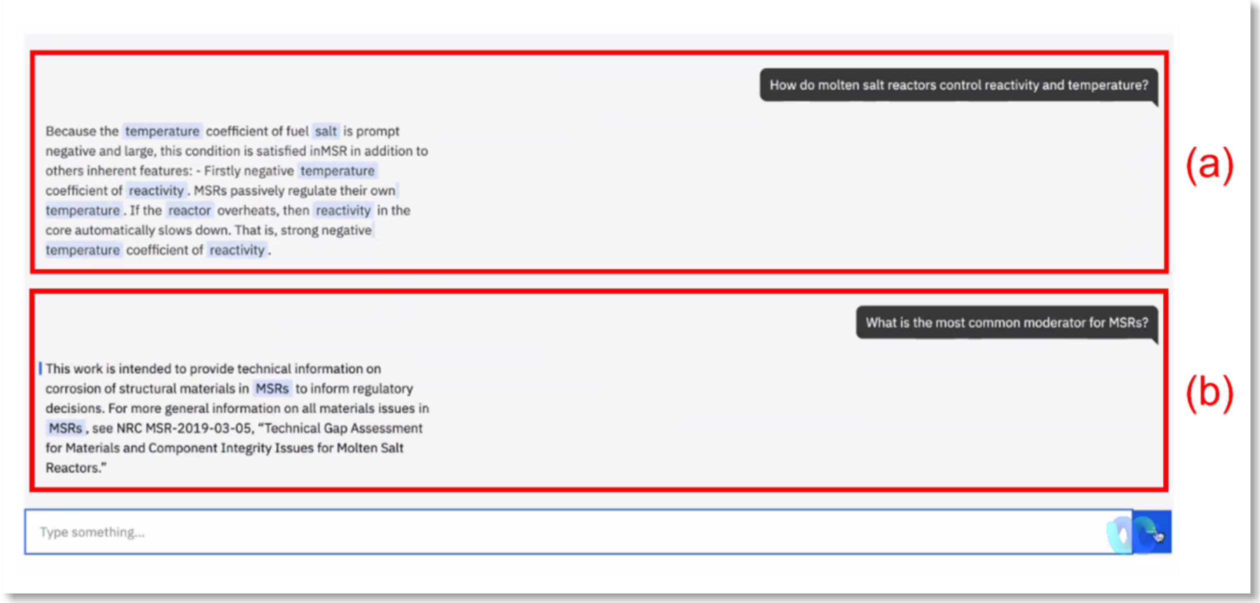
The common objective of the MOSART and MSFR projects is to develop a conceptual design for intermediate/fast- spectrum MSRs with an effective system configuration e result- ing from physical, chemical and material studies e for the reactor core, the reprocessing unit and waste conditioning (Merle-Lucotte et al., 2013).

Economics and finance of Molten Salt Reactors

Siting and licensing costs, site labour and common facilities mostly drive such cost reduction. Therefore, two identical MSRs at the same site are envisaged to cost less than doubling the cost of a single MSR. 2.6.3. Modularisation

Figure 2-7 Example relevancy training in Vendor 2's cloud service

After some initial relevancy training was completed for a selection of questions, Vendor 2's team created an environment using the chatbot to allow the RES staff to test the answering capability of the service. In this environment, the user poses a question in the entry box and the chatbot answers with an excerpt from the library of accessible data. In a meeting with RES staff, Vendor 2's team operated this test environment and demonstrated its preliminary capabilities. Figure 2-8 shows two representative responses from the service.



How do molten salt reactors control reactivity and temperature?

Because the temperature coefficient of fuel salt is prompt negative and large, this condition is satisfied inMSR in addition to others inherent features: - Firstly negative temperature coefficient of reactivity. MSRs passively regulate their own temperature. If the reactor overheats, then reactivity in the core automatically slows down. That is, strong negative temperature coefficient of reactivity.

What is the most common moderator for MSRs?

This work is intended to provide technical information on corrosion of structural materials in MSRs to inform regulatory decisions. For more general information on all materials issues in MSRs, see NRC MSR-2019-03-05, "Technical Gap Assessment for Materials and Component Integrity Issues for Molten Salt Reactors."

Figure 2-8 Vendor 2 RESbot responses to the questions, (a) "How do molten salt reactors control reactivity and temperature?" and (b) "What is the most common moderator for MSRs?"

The RES staff qualitatively evaluated the returned results from Vendor 2's service. The answer to the question, "How do molten salt reactors control reactivity and temperature?" (shown in Figure 2-8a) was judged to be reasonable and accurate. However, the answer to the question, "What is the most common moderator for MSRs?" (shown in Figure 2-8b) was judged not relevant or applicable. The answer from Figure 2-8a had the keywords "salt", "reactor", “reactivity”, and “temperature,” all of which were in the question. The answer from Figure 2-8b had the term “MSRs,” but it did not include the keyword or concept “moderator.” Afterwards, the RES staff performed additional relevancy training for the Figure 2-8b question by rating more passages. It received the same answer, so more extensive training of the model would be necessary to improve the results using Vendor 2’s cloud service.

2.4 Summary

An autonomous RESbot researcher in knowledge mining applications in the nuclear domain requires significant training due to the unique terminology. An investment of staff time and resources to train the NLP models was required for both the commercial services investigated here.

Vendor 1’s custom NER tool, which the RES staff did not have the resources to properly train and activate for this use case, is well-suited for repeated applications and entities (e.g., forms, references to regulations). Vendor 1’s generic search service delivered reasonable results on some questions; however, training and customization would be required to consistently deliver quality answers. Vendor 2 established a highly functional and intuitive user interface, a useful summarization capability, and established a relevancy training process that is simple and easy to perform. Like with the first vendor, Vendor 2’s cloud service returned some accurate responses, and some of the retrieved results were irrelevant. Some points to consider with Vendor 2’s approach are that relevancy training needs to be conducted for each question a user would potentially ask, and that relevancy training could be an ongoing process where users rate the relevancy of the responses.

There is accelerated progress in developing AI/ML solutions for NLP in the commercial sector, such that it is likely that the type of RESbot envisioned for this project may be fully functional within 5 years. For instance, OpenAI [7] has developed an AI/ML NLP tool called “GPT-3” that can be used for many purposes, including content generation, summarization, data extraction, and classification. The OpenAI models can be fine-tuned with additional training data, and they may be worth exploring in future work.

3 USE CASE 2 INVESTIGATION—INTELLIGENT SEARCH

3.1 Background

The objective of Use Case 2 of the RESbot project was to develop an autonomous researcher for intelligent search of the problem space to support modeling and simulation applications.

3.1.1 Regulatory Context

As highlighted in NRC Regulatory Guide (RG) 1.245, “Preparing Probabilistic Fracture Mechanics Submittals,” Revision 0, issued January 2002 [7], the NRC staff has recently observed an increase in the number of applications using PFM as a technical basis. The heightened focus on PFM is partly due to the increased emphasis on risk-informed regulation, but also because plant aging and new degradation mechanisms can be difficult to address using traditionally conservative deterministic fracture mechanics. The increased use of PFM has also been facilitated by improvements in computational capabilities and the increased availability of PFM codes such as xLPR, which is described in NUREG-2247, “Extremely Low Probability of Rupture Version 2 Probabilistic Fracture Mechanics Code,” issued August 2021 [8]. Furthermore, the NRC staff has used PFM methods to inform regulatory activities, such as in its assessment of RG 1.99, “Radiation Embrittlement of Reactor Vessel Materials,” Revision 2, issued May 1988 [9], and its assessment of the effects of primary water stress corrosion cracking (PWSCC) in pressurized-water reactor piping systems previously approved for leak-before-break consistent with the requirements of Title 10 of the *Code of Federal Regulations*, Part 50, Appendix A, Criterion 4 (GDC 4) [10]. The RG 1.99 assessment is documented TLR-RES/DE/CIB-2020-09, “RG 1.99 Revision 2 Update FAVOR Scoping Study,” issued October 2020 [11]; the GDC 4 assessment is documented in TLR-RES/DE/REB-2021-14-R1, “Probabilistic Leak-Before-Break Evaluations of Pressurized-Water Reactor Piping Systems using the Extremely Low Probability of Rupture Code,” issued April 2022 [12]. NUREG/CR-7278, “Technical Basis for the use of Probabilistic Fracture Mechanics in Regulatory Applications,” issued January 2022 [13], constitutes the technical basis for RG 1.245 and develops the concept of a PFM analysis methodology and outlines important considerations for a high-quality and high-confidence PFM analysis.

3.1.2 PFM Technology

Fracture mechanics is concerned with the basic methods for predicting the load-carrying capabilities of components containing cracks [14]. Historically, fracture mechanics analyses have most commonly been performed deterministically. However, to understand more completely the various problem uncertainties, stochastic fracture mechanics analyses should be performed. This stochastic analysis is the field of application of PFM codes.

Figure 3-1 illustrates a simplistic PFM analysis. The curve on the left represents the distribution of crack-driving force or applied stress-intensity factor (SIF), which depends on the uncertainties in stress and crack size. The curve on the right represents the toughness distribution or critical (i.e., allowable) SIF of the material. When the two distributions overlap, there is a finite probability of failure, which is indicated by the shaded area. Time-dependent crack growth, such as from fatigue or stress-corrosion cracking or both, can be considered by applying the appropriate growth laws to the crack distribution. Crack growth can cause the applied SIF distribution to shift to the right with time, thereby increasing the probability of failure.

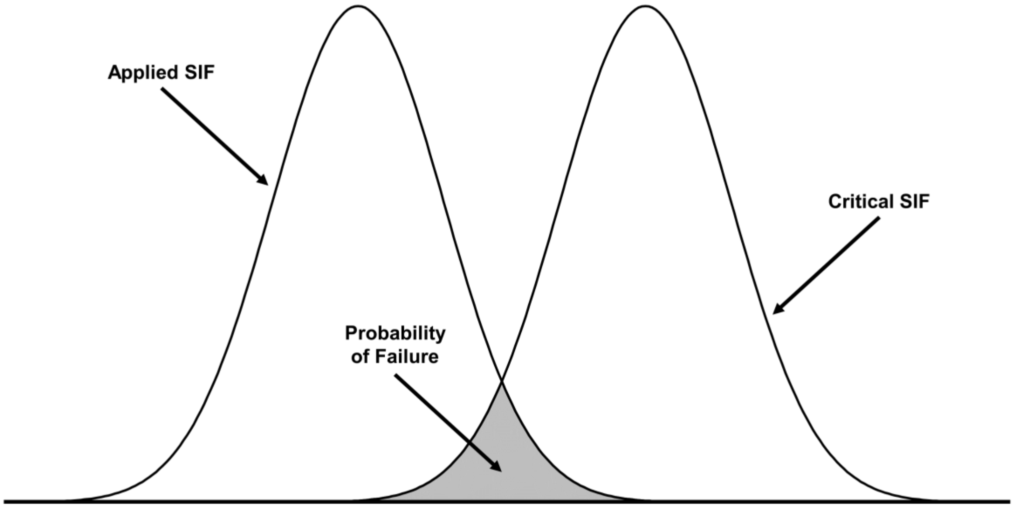


Figure 3-1 Illustration of PFM analysis

Source: Adapted from [15], Fig. 9.39

As described in the 2005 *Fracture Mechanics Fundamentals and Applications* [15], the overlap of the two probability distributions shown in Figure 3-1 represents a simple case. In most practical situations, however, there is randomness or uncertainty associated with many variables. Monte Carlo simulation can estimate failure probability in such cases by propagating input values, sampled from distributions representing the uncertainty in those variables, through a deterministic model using numerous “trials.” The results can then be analyzed using statistical methods to assess the risk of such failures occurring. This is the basic approach used in the xLPR code.

xLPR is a PFM code for piping applications. The code was developed jointly by the RES and the Electric Power Research Institute. They engaged in a multiyear code development effort that built on the results of a successful pilot study. The code was designed, programmed, and tested under a rigorous software quality assurance program, and provides regulators, industry, researchers, and the public with the capabilities to quantitatively analyze the risks associated with nuclear power plant piping systems subject to active degradation mechanisms. Some core capabilities of the code include modeling stress corrosion cracking (e.g., PWSCC), effects of welding residual stresses (WRS), leak rates, and rupture. The xLPR code also supports many probabilistic features, such as a wide range of probability distributions for inputs and advanced sampling techniques like importance sampling.

3.1.3 PFM Analysis

To maximize the effectiveness and efficiency of NRC staff reviews of licensing submittals, RG 1.245 and NUREG/CR-7278 outline five basic analytical steps for preparing a PFM analysis. These steps are the following:

1. translate regulatory requirements into an analysis plan
2. characterize model input uncertainty
3. estimate quantities of interest and their associated uncertainty, which includes assessment of sampling uncertainty, sensitivity analysis, and output uncertainty analysis
4. conduct sensitivity studies to assess the credibility of modeling assumptions
5. draw conclusions from analysis results

The process is iterative in that the analysis results are synthesized to refine the analysis until a conclusion is drawn.

As stated in NUREG/CR-7278 Section 3.3.3, sensitivity analysis focuses on identifying how the input uncertainties contribute to the uncertainty in the output quantities of interest (QoIs). Sensitivity analyses help to identify problem drivers, which are defined as uncertain model inputs that explain substantial uncertainty in the model output. Understanding the problem drivers allows the analyst to do several things. The first is to confirm that the model is behaving as expected. Understanding the problem drivers also allows the analyst to identify inputs whose uncertainty distributions are themselves uncertain and that may need refinement before final estimation of the QoI. In addition, it allows the analyst to identify assumptions that are uncertain and thus may be candidates for sensitivity studies. Further, understanding the problem drivers improves the accuracy of the output uncertainty analysis by reducing the dimension of the input space and identifying important inputs that can be used in more targeted sampling methods, such as importance sampling. Sensitivity analysis plays a critical role in improving output uncertainty analysis. A common goal of a PFM analysis is to accurately estimate a QoI along with its associated uncertainty. By informing the final sampling scheme, sensitivity analyses can improve QoI estimation.

As stated in NUREG/CR-7278 Section 3.3.3, sensitivity studies are supplemental analyses conducted under different, yet plausible, assumptions. Their purpose is to challenge uncertain analysis assumptions that could substantively change the analysis results. The goal is to conduct enough sensitivity studies such that there is a sufficiently low chance that the results of the analysis depend heavily on unverifiable or uncertain assumptions. Uncertain analysis assumptions can be classified as either modeling assumptions or input parameter specification assumptions. Modeling assumptions include any assumptions in the computational modeling framework, while input parameter specification assumptions refer to any assumptions made when specifying the values of the input parameters to the PFM model. An example of a common type of sensitivity study includes considering changes in the results if a different probability distribution for an uncertain input (or several uncertain inputs) is used.

3.1.4 ML Technology

According to the 2016 *Deep Learning* [16], ML is an approach for improving the performance of an algorithm through consideration of data and associated characteristic features pertinent to a specific problem domain. ML can be motivated (a) purely statistically (e.g., by imputing and leveraging statistical relationships amongst and across features), (b) biologically (e.g., to model and understand how biological agents, such as humans, learn and form expertise in specific subjects), (c) computationally (e.g., how can an algorithm improve its performance over time), or (d) in a variety of other ways, including self-monitored adaptive control (e.g., via reinforcement learning). For Use Case 2, ML can be beneficial by automating or semi-automating the difficult task of analyzing streams of data, such as when conducting a sensitivity analysis. Such an application is where data-driven algorithms, such as ML models, can provide support for human analysts. In this case, the data-driven algorithms can provide an efficient and easy-to-use pipeline from the raw data to specific sensitivity analysis results of interest to a human analyst.

There are several kinds of ML algorithms, such as unsupervised, supervised, and reward-based. Unsupervised algorithms, such as clustering, are used to find structure in data without further (supervisory) information from the human analyst. Supervised learning provides a mechanism for generating a mapping (or function) from input features (describing a particular domain phenomena) to target outputs (or QoIs to the human analyst). For the Use Case 2 investigations, supervised learning algorithms were used to facilitate efficient sensitivity analysis and provide support for sensitivity studies.

3.2 Approach

The NRC staff approached its investigations for Use Case 2 as follows:

- problem selection
- PFM code interface development
- ML analysis development

Section 3.2.1 describes problem selection. Sections 3.2.2 and 3.2.3 detail development of the PFM code interface and ML analysis, respectively. Development of these two aspects entailed coding new software modules using open-source software. Together, they were called “xLPRbot” and used to automate sensitivity analysis and sensitivity studies using the xLPR code. An SME from the RES staff developed the PFM code interface module, and Sandia National Laboratories (SNL) developed the ML analysis module under contract to the NRC. The development activities were conducted in concert. Figure 3-2 illustrates the architecture of xLPRbot.

Both the RES staff and SNL chose Python as the computer coding language for this effort. Python was chosen for its overall ease of use, flexibility, and because it is particularly effective for data science and analytics applications. With respect to its ease of use, the user need only find and install the appropriate Python package to access the desired functionality. For example, scikit-learn, as described by Pedregosa, et al. [17], provides advanced algorithms leveraging existing data science, ML, and AI algorithms. Other Python packages provide easy access to statistical ML, such as linear regression models and supervised ML, including random forest regression models, which are described by Breiman [18], and deep neural networks, which are described by Hinton, et al. [19]. Other relevant tools that Python packages provide access to are feature importance, as described at scikit-learn.org [20], and confidence interval (CI) bootstrapping. Python also offers quick access to many data file formats, such as Java Script object notation (JSON), and various visualization packages, such as Matplotlib described at matplotlib.org [21].

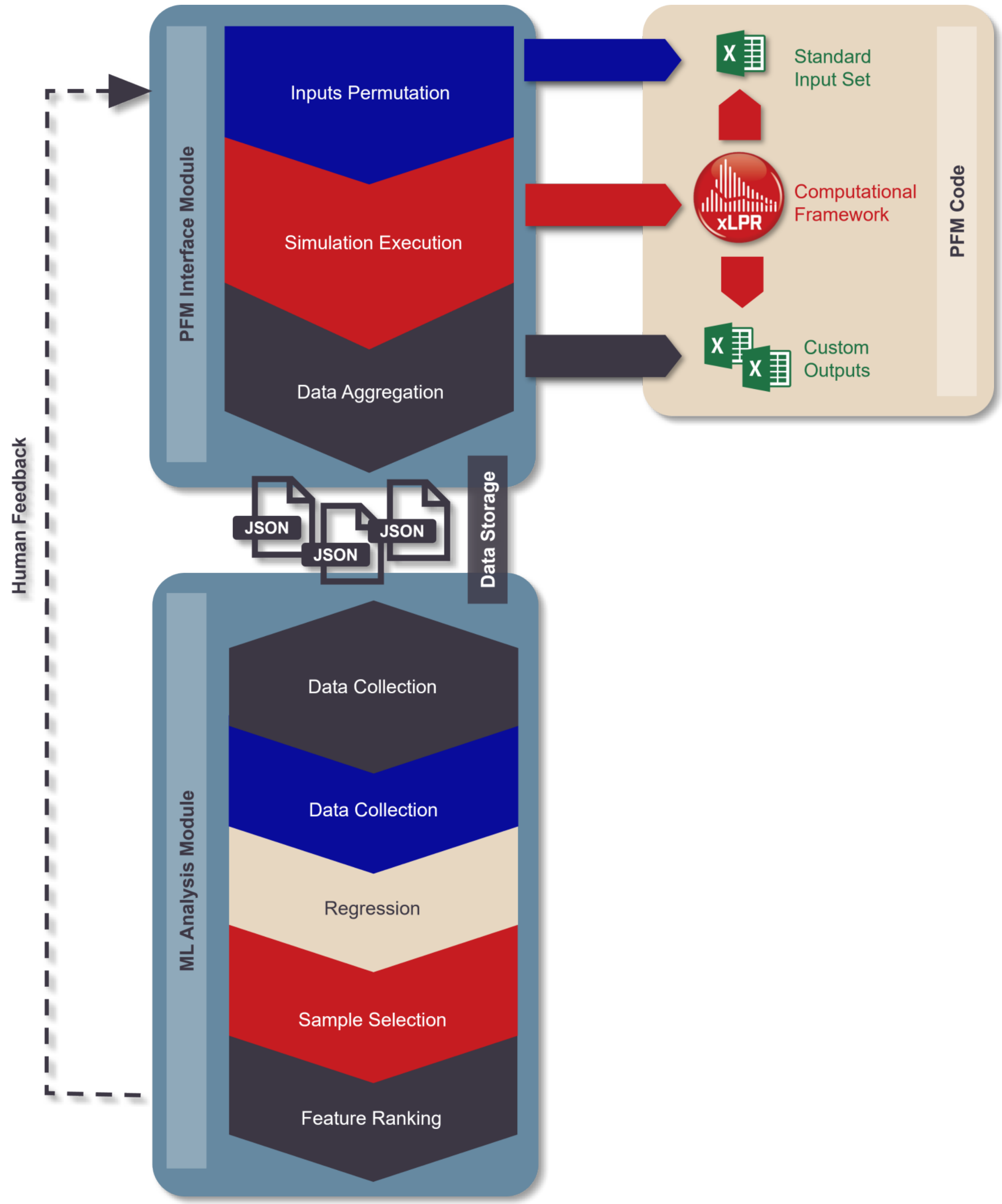


Figure 3-2 xLPRbot architecture

3.2.1 Problem Selection

For the PFM analysis, the NRC staff selected one of probabilistic leak-before-break analysis cases from TLR-RES/DE/REB-2021-14-R1. Specifically, Case 2.1.1 analyzes the behavior of an un-mitigated, Westinghouse-designed, pressurizer surge line piping to pressurizer nozzle dissimilar metal weld. This case focuses on pre-existing cracks that are subject to PWSCC growth. For the Use Case 2 explorations, the Case 2.1.1 was simplified somewhat to focus on a reduced scope of data. For instance, the analysis was limited to a single circumferential crack and the simulated plant operation time was reduced to 240 months. The normal operating stresses were also converted to loads to study their potential influence. The effects of seismic events, leak rate detection, and inservice inspection were also omitted.

The following QoIs were selected for analysis, where the information in parentheses represents the assigned variable names that correspond with the results presented in Section 3.3:

1. occurrence of leak (is_leaking)
2. occurrence of rupture (is_ruptured)
3. total leak rate in kilograms per second (total_leak_rate)
4. normalized circumferential crack depth (cc_depth_normalized)
5. normalized circumferential inside diameter crack length (cc_ID_length_normalized)
6. normalized circumferential outside diameter crack length (cc_OD_length_normalized)

These QoIs were selected because they describe key aspects of the predicted component behavior (i.e., whether it leaks or ruptures). They were also selected because they represent different data types. For instance, the occurrence of rupture is a binary output (equal to 0 if no rupture has occurred and equal to 1 when a rupture has occurred), the total leak rate is a semicontinuous output (either equal to 0 or a distributed range of values), and the crack dimensions are all continuous outputs (distributed ranges of values).

3.2.2 PFM Code Interface Development

The PFM code interface module comprises some 570 source lines of code in Python. As shown in the Figure 3-2, the module has three primary routines:

1. inputs permutation
2. simulation execution
3. data aggregation

The inputs permutation routine interfaces with the xLPR input set, which is a Microsoft Excel file. It parses the input set and, to support the sensitivity studies described in Section 3.3.3, it can modify certain input parameters and push the changes back to the input set. It uses functions from the Python standard library and the open-source Pandas and openpyxl libraries.

The simulation execution routine interfaces with the xLPR computational framework, which is implemented in a GoldSim model file. The primary role of this routine is to automatically execute the xLPR code. It includes a feature for parallel, looped, or both parallel and looped execution of multiple GoldSim instances to increase the total amount of data that can be stored while also decreasing the overall simulation runtime. Indeed, GoldSim includes native parallelization capabilities though its distributed processing module; however, use of this module does not support extraction of the input sample and results data, which is requirement for the ML analysis module. The simulation execution routine uses functions from the Python standard library.

The xLPR computational framework was modified to save the input samples and time history data for the selected QOIs to a collection of Microsoft Excel files. The framework operates by drawing samples for every input distribution, regardless of whether that input impacts the analysis. Subject matter expert review, which included examination of the framework logic, was used to determine the set of 56 input variables that are germane to this analysis. Table 3-1 lists these inputs. Custom variable names were assigned because they are more descriptive than those used within the xLPR code. For traceability, the xLPR input set global identification (ID) numbers are provided.

Table 3-1 Relevant input variables

Deterministic Input Custom Variable Name (xLPR Global ID No.)	Probabilistic Input Custom Variable Name, Distribution Type (xLPR Global ID No.)
pipe_outside_diameter (Global ID 1101)	initial_cc_full_length, lognormal (Global ID 1210)
pipe_wall_thickness (Global ID 1102)	initial_cc_depth, lognormal (Global ID 1212)
initial_cc_full_length_multiplier (Global ID 1211)	WRS_axial_premitigation_pt01 through WRS_axial_premitigation_pt26, normal (Global ID 4352)
initial_cc_depth_multiplier (Global ID 1213)	left_pipe_material_yield_strength, lognormal (Global ID 2101)
hydrogen_concentration_initial (Global ID 3002)	left_pipe_material_ultimate_strength, lognormal (Global ID 2102)

Deterministic Input Custom Variable Name (xLPR Global ID No.)	Probabilistic Input Custom Variable Name, Distribution Type (xLPR Global ID No.)
operating_pressure_period_1 (Global ID 3101)	left_pipe_material_elastic_modulus, normal (Global ID 2105)
operating_temperature_period_1 (Global ID 3102)	right_pipe_material_yield_strength, lognormal (Global ID 2301)
force_along_x_axis_normal_thermal_ expansion_period_1 (Global ID 4105)	right_pipe_material_ultimate_strength, lognormal (Global ID 2302)
moment_about_y_axis_normal_thermal_ expansion_period_1 (Global ID 4107)	right_pipe_material_elastic_modulus, normal (Global ID 2305)
weld_material_PWSCC_growth_power_law_ constant_alpha (Global ID 2588)	weld_material_fracture_toughness_JIc, normal (Global ID 2506)
weld_material_PWSCC_growth_power_law_ exponent_beta (Global ID 2589)	weld_material_fracture_toughness_ coefficient_C, normal (Global ID 2507)
weld_material_PWSCC_growth_stress_ intensity_factor_threshold_Kth (Global ID)	weld_material_fracture_toughness_ exponent_m, normal (Global ID 2508)
weld_material_PWSCC_growth_ factor_of_improvement (Global ID 2596)	weld_material_PWSCC_growth_ activation_energy_Qg, normal (Global ID 2591)
weld_material_PWSCC_growth_ reference_temperature (Global ID 2597)	weld_material_PWSCC_growth_ component_to_component_ variability_factor_fcomp, lognormal (Global ID 2592)
	weld_material_PWSCC_growth_ within_component_variability_factor_fflaw, lognormal (Global ID 2593)

Deterministic Input Custom Variable Name (xLPR Global ID No.)	Probabilistic Input Custom Variable Name, Distribution Type (xLPR Global ID No.)
	weld_material_PWSCC_growth_peak_to_valley_ECP_ratio_minus1_P-1, lognormal (Global ID 2594)
	weld_material_PWSCC_growth_characteristic_peak_width_vs_ECP_c, normal (Global ID 2595)

The data aggregation function interfaces with the input sample and results data saved by the xLPR computational framework to a collection of Microsoft Excel files. It parses the data in these files and, if parallel or looped simulation execution was performed, assembles the data from multiple files into single datasets. It then saves the data in JSON files. JSON is a lightweight data-interchange format, and these files serve as the input for the ML analysis module. The data aggregation routine uses functions from the Python standard library and the Pandas library. The developers opted for the JSON data storage approach because it was not practical to implement a direct software link between the PFM code interface and ML analysis modules. In future iterations of xLPRbot, the two modules could be integrated, and the necessary information could just be saved to a convenient data structure internal to the code, such as a Pandas DataFrame.

3.2.3 ML Analysis Development

The ML analysis module comprises some 360 source lines of code in Python. As shown in the Figure 3-2, the module has four primary routines:

1. data collection
2. regression
3. sample selection
4. feature ranking

The data collection routine brings the files containing the xLPR code inputs and output QoIs together into a format for use in supervised ML. In supervised ML, the set of inputs comprises the features, and the QoIs are the targets.

In the regression routine, a regression is performed using a random forest. A random forest learner is composed of multiple decision-tree learners operating in tandem (i.e., in an ensemble). For regression of real-valued QoIs, the output of each individual decision-tree learner is aggregated using the average to provide the final output prediction. The regression can be performed across all QoIs simultaneously (i.e., multivariate analysis) or for each QoI individually (i.e., univariate analysis).

The sample selection routine computes the 5th and 95th binomial confidence bounds via a bootstrapping approach. These bounds can be used in successive requests to the PFM code interface module for more data, typically by increasing the requested number of random xLPR code realizations by orders of magnitude. When the ML analysis module determines that more realizations are necessary, it requests more data form the PFM code interface module via the feedback loop.

When enough samples have been generated, the feature ranking routine determines the importance values for each xLPR code input in relation to a specific QoI in the case univariate analysis, or across all QoIs simultaneously in the case of multivariate analysis. Once the importance of all the inputs has been ranked, the results can then be used to support sensitivity analyses and sensitivity studies consistent with RG 1.245.

For the Use Case 2 investigations, xLPRbot was not fully automated because the feedback loop relied on a human analyst to interpret the data from the ML analysis module (e.g., to decide when enough samples had been generated) and to make decisions using that data to re-execute the XLPR code via the PFM code interface module. However, after establishing numerical measures for convergence and input importance, it would be possible to fully automate the bot.

3.3 Results and Discussion

Following a convergence analysis presented in Section 3.3.1, xLPRbot was used to automate a sensitivity analysis and sensitivity studies for a PFM analysis using the xLPR code. The sensitivity analysis and sensitivity study results are presented in Sections 3.3.1 and 3.3.3, respectively.

3.3.1 Convergence Analysis

The first step was to determine the appropriate number of randomly generated samples for characterizing the phenomena of interest, or QoIs, from the PFM model (i.e., the xLPR code) as specified using input distributions. To determine the appropriate number of samples or realizations needed for the sensitivity analysis, SNL used the bootstrapping method to compute the 95 percent CIs for each QoI for successively larger sample sizes. Specifically, SNL requested successively larger sample sizes of 200, 2,000 and 20,000 realizations from the xLPR code. SNL then computed the 95 percent CIs for each QoI to see when they become sufficiently tight around the mean value. This analysis is shown in Figure 3-3 for the binary-valued “is_ruptured” QoI and in Figure 3-4 for the continuously distributed “total_leak_rate” QoI.

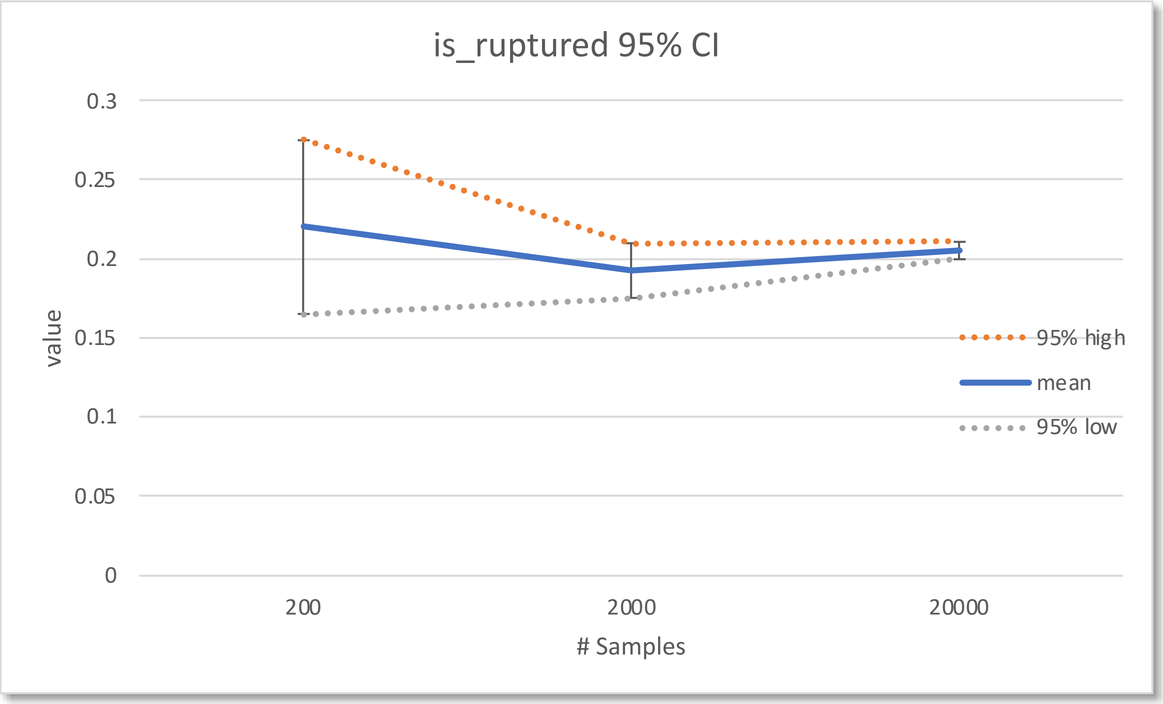


Figure 3-3 CIs for the "is_ruptured" QoI from 200, 2,000, and 20,000 realizations

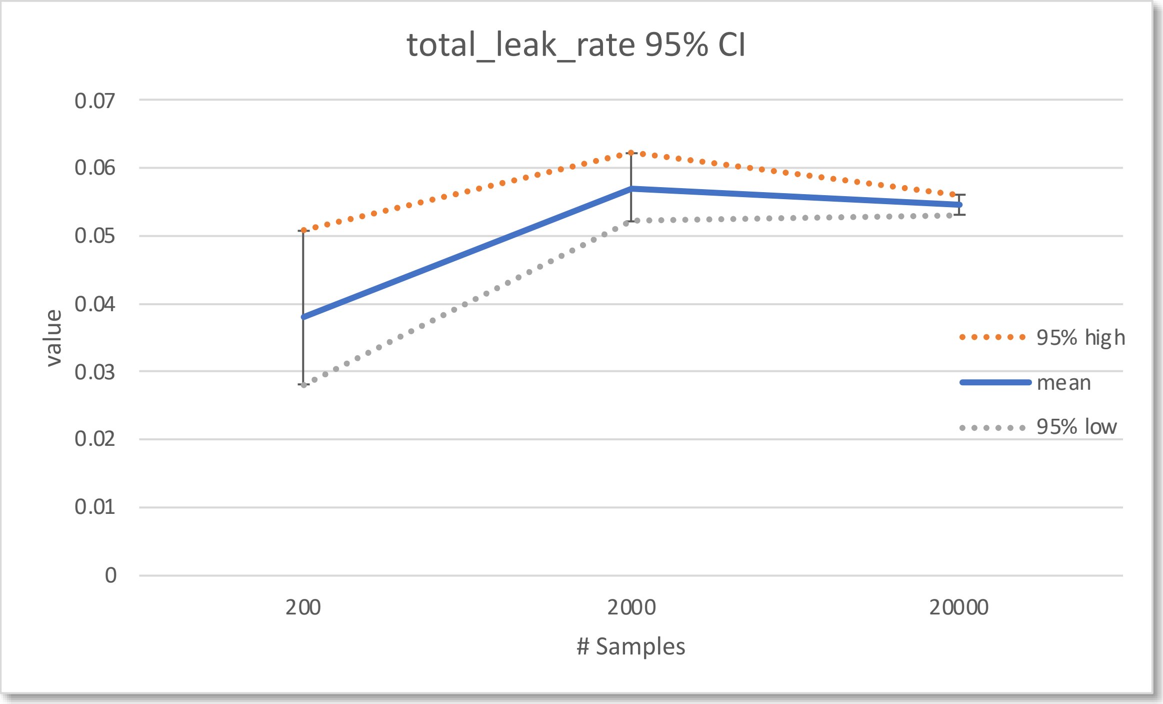


Figure 3-4 CIs for the "total_leak_rate" QoI from 200, 2,000, and 20,000 realizations

In both Figure 3-3 and Figure 3-4, 2,000 random realizations seem to be enough. This observation is supported by Figure 3-4, for example, where 200 random realizations is not sufficient because the CIs do not overlap with the CIs for 2,000 random realizations, whereas the CIs for 2,000 random realizations do overlap with the CIs for 20,000 random realizations. Note that for both the “cc_OD_length_normalized” (not shown) and “total_leak_rate” QoIs, the CIs were computed using only those samples where there were non-zero values to mitigate the bias from those cases where there was no outside diameter crack dimension or no leak, respectively. In the end, SNL chose to use 2,000 samples for further sensitivity analysis.

3.3.2 Sensitivity Analysis

After the convergence analysis was completed, the next step was finding those xLPR code input variables that are most important with respect to the QoIs, both individually (univariate analysis) and across all the QoIs (multivariate analysis). SNL used supervised ML, in the form of random forest regression, to find the most important inputs. Supervised learning creates a mapping from inputs to outputs. In this case, the RES staff used the PFM code interface module to provide SNL with both the xLPR code inputs and outputs in the form of JSON files. SNL then asked the random forest regressor, via the ML analysis module, to learn a mapping between the inputs and outputs. In some sense, supervised ML is like function approximation, and thus the random forest regressor learns a function (or functional mapping) from inputs to the outputs. The random forest regression method utilizes ensemble learning, which means that multiple ML models are learned in tandem and the average output value across the entire ensemble is reported as the predicted (or computed) output. Thus, in this instance, the random forest regressor is a specific kind of surrogate model for the xLPR code. Note that the use of ML for sensitivity analysis can be achieved without any further human intervention. Once the ML model is trained, there are several ways to determine the most important inputs. The random forest ML method and its scikit-learn implementation include a function to compute and rank the most important input features as measured by the mean decrease in impurity, which is described in the 1984 *Classification and Regression Trees* [22]. This measure works well for classification tasks, but since the subject problem is that of regression, SNL also employed permutation importance, which is described in Altmann, et al. [23]. Permutation importance can be computed using any trained, supervised ML model along with the inputs and outputs used during training, including linear regression. Permutation importance requires longer run-times, however.

3.3.2.1 Univariate Analysis

The univariate permutation importance results for the top ten most important input variables are shown in Table 3-2 for the “is_ruptured” QoI and in Table 3-3 for the “total_leak_rate” QoI. As can be seen from these tables, the input variable, “WRS_axial_premitigation_pt01,” has an outsize importance as compared to all other input variables. These results make sense because this variable represents the WRS on the inside diameter of the pipe, and its contribution to the applied stresses are needed to begin to grow the crack that was initially seeded in the analysis. For consistency and illustrative purposes, only the results for the same QoIs from Section 3.3.1 are presented; however, except for the “total_leak_rate” QoI, the top five most important inputs were the same for all the QoIs.

Table 3-2 Top ten input permutation importance values for the “is_ruptured” QoI using random forest regression model

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.9036
weld_material_PWSCC_growth_component_to_component_variability_factor_fcomp	0.2051
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.1420
WRS_axial_premitigation_pt14	0.0397
WRS_axial_premitigation_pt26	0.0270
WRS_axial_premitigation_pt22	0.0249
WRS_axial_premitigation_pt23	0.0165
WRS_axial_premitigation_pt24	0.0158
WRS_axial_premitigation_pt02	0.0143
WRS_axial_premitigation_pt07	0.0137

Table 3-3 Top ten input permutation importance values for the “total_leak_rate” QoI using random forest regression model

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	1.2596
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.2575
WRS_axial_premitigation_pt26	0.0437
WRS_axial_premitigation_pt22	0.0395
left_pipe_material_elastic_modulus	0.0387
WRS_axial_premitigation_pt20	0.0369
WRS_axial_premitigation_pt10	0.0340
weld_material_PWSCC_growth_component_to_component_variability_factor_fcomp	0.0332
WRS_axial_premitigation_pt21	0.0249
WRS_axial_premitigation_pt17	0.0220

3.3.2.2 Multivariate Analysis

One benefit of using supervised ML in this research is to perform multivariate analyses as easily as univariate ones. The multivariate permutation importance results for the top ten most important input variables are shown in Table 3-4 for all the QoIs. Again, “WRS_axial_premitigation_pt01” is the most important, and the top five most important inputs are consistent with the univariate results from Section 3.3.2.1.

Table 3-4 Top ten input permutation importance values for all QoIs using random forest regression model

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.9719
weld_material_PWSCC_growth_component_to_component_variability_ factor_fcomp	0.2024
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.1497
WRS_axial_premitigation_pt14	0.0306
WRS_axial_premitigation_pt26	0.0275
WRS_axial_premitigation_pt22	0.0230
WRS_axial_premitigation_pt02	0.0175
WRS_axial_premitigation_pt24	0.0163
WRS_axial_premitigation_pt23	0.0161
WRS_axial_premitigation_pt07	0.0160

3.3.2.3 Validation

To validate the ML-based solution, SNL compared the results to those generated using simple linear regression. The permutation importance results using linear regression for the top ten most important input variables are shown in Table 3-5 for the “is_ruptured” QoI and in Table 3-6 for the “total_leak_rate” QoI. The results using linear regression are comparable to the results using ML as presented in Table 3-2 and Table 3-3. Of note, the linear regression model cannot support multivariate analysis like the random forest regression model can. This additional capability is viewed as a key improvement over prior sensitivity analysis approaches that largely relied on linear regression, for example, in SAND2017-2854, “xLPR Scenario Analysis Report,” issued March 2017 [24], and TLR-RES/DE/CIB-2021-11, “Sensitivity Studies and Analyses Involving the Extremely Low Probability of Rupture Code,” issued May 2021 [25]. Additionally, SNL expects that, for highly non-linear systems, the non-linear potential for a supervised ML model would be a benefit over linear regression.

Table 3-5 Top ten input permutation importance values for the “is_ruptured” QoI using linear regression model

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.6535
weld_material_PWSCC_growth_component_to_component_variability_factor_fcomp	0.0597
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.0356
WRS_axial_premitigation_pt26	0.0193
WRS_axial_premitigation_pt09	0.0054
WRS_axial_premitigation_pt02	0.0046
WRS_axial_premitigation_pt08	0.0045
WRS_axial_premitigation_pt20	0.0041
WRS_axial_premitigation_pt22	0.0033
WRS_axial_premitigation_pt12	0.0032

Table 3-6 Top ten input permutation importance values for the “total_leak_rate” QoI using linear regression model

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.4827
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.0311
WRS_axial_premitigation_pt23	0.0127
WRS_axial_premitigation_pt26	0.0108
left_pipe_material_yield_strength	0.0079
right_pipe_material_yield_strength	0.0079
WRS_axial_premitigation_pt12	0.0063
WRS_axial_premitigation_pt20	0.0061
WRS_axial_premitigation_pt10	0.0059
WRS_axial_premitigation_pt21	0.0047

3.3.3 Sensitivity Studies

3.3.3.1 Effect of Converting Deterministic Inputs to Probability Distributions

As stated in NUREG/CR-7278 Section 3.2.1, understanding the rationale for classifying inputs as deterministic or uncertain is important when interpreting the analysis results. Deterministic inputs may be fixed to single values for several reasons, including (a) they have a known physical value (e.g., a known yield strength of a material), (b) the chosen fixed value is determined to be a value of interest (e.g., a conservative value used for a specific reason or a value of relevance for sensitivity studies), or (c) including uncertainty would not affect decision-making. Data, expert judgment, and sensitivity analysis inform whether an input should be modeled as deterministic or uncertain.

In the analysis of the selected problem, several inputs were specified as deterministic. Thus, a sensitivity study was conducted using ML to assess whether these inputs should instead be modeled as uncertain. For this sensitivity study, the inputs permutation routine was invoked to automatically convert the following deterministic variables to probability distributions:

- pipe_outside_diameter
- pipe_wall_thickness
- initial_cc_full_length_multiplier
- initial_cc_depth_multiplier
- hydrogen_concentration_initial
- operating_pressure_period_1
- operating_temperature_period_1
- force_along_x_axis_normal_thermal_expansion_period_1
- moment_about_y_axis_normal_thermal_expansion_period_1

To produce the new distributions, a normal distribution was assumed with the mean set equal to the deterministic value and the standard deviation set to 10 percent of the mean. Additionally, upper and lower truncation points were set at one standard deviation above and below mean.

With these updates, the univariate permutation importance results using ML for the top ten most important input variables are shown in Table 3-7 for the “is_ruptured” QoI and Table 3-8 for the “total_leak_rate” QoI, and the multivariate results are shown in the Table 3-9. The most important input variable, “WRS_axial_premitigation_pt01,” remains the same; however, significant differences can be seen in the rest of the input rankings. For example, the variable “operating_temperature_period_1” now appears in the top three most important inputs, whereas, before this input wasn’t even in the top ten. This variable represents the temperature of the reactor coolant fluid inside the pipe, and given the role of temperature in the PWSCC growth model, it’s reasonable to expect this value to have an influence on the results. It’s noted that uncertainty was applied to this input variable arbitrarily just because it has been previously classified as a deterministic input. However, the operating temperate is tightly controlled in the reactor coolant system and it is not expected to experience the degree of variation that was modeled in this sensitivity study. Thus, sensitivity changes like these are possible to study using ML methods as shown here, but this result in particular highlights the need for judicious application on the part of the human analyst.

Table 3-7 Top ten input permutation importance values for the "is_ruptured" QoI using random forest regression model after converting deterministic inputs to probability distributions

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.7418
operating_temperature_period_1	0.4228
pipe_outside_diameter	0.1387
weld_material_PWSCC_growth_component_to_component_variability_factor_fcomp	0.0536
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.0243
pipe_wall_thickness	0.0220
WRS_axial_premitigation_pt24	0.0218
WRS_axial_premitigation_pt25	0.0206
WRS_axial_premitigation_pt02	0.0192
WRS_axial_premitigation_pt19	0.0157

Table 3-8 Top ten input permutation importance values for the "total_leak_rate" QoI using random forest regression model after converting deterministic inputs to probability distributions

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.7916
WRS_axial_premitigation_pt22	0.1303
operating_temperature_period_1	0.1045
WRS_axial_premitigation_pt25	0.0559
WRS_axial_premitigation_pt15	0.0498
WRS_axial_premitigation_pt14	0.0460
weld_material_PWSCC_growth_activation_energy_Qg	0.0381

Probabilistic Input	Permutation Importance Value
pipe_outside_diameter	0.0333
WRS_axial_premitigation_pt26	0.0277
WRS_axial_premitigation_pt07	0.0272

Table 3-9 Top ten input permutation importance values for all QoIs using random forest regression model after converting deterministic inputs to probability distributions

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt01	0.7545
operating_temperature_period_1	0.4354
pipe_outside_diameter	0.1219
weld_material_PWSCC_growth_component_to_component_variability_ factor_fcomp	0.0750
pipe_wall_thickness	0.0262
WRS_axial_premitigation_pt24	0.0222
WRS_axial_premitigation_pt25	0.0219
WRS_axial_premitigation_pt02	0.0213
weld_material_PWSCC_growth_within_component_variability_factor_fflaw	0.0208
WRS_axial_premitigation_pt15	0.0206

3.3.3.2 Effect of Changing the Probability Distribution Tails

As described in NUREG/CR-7278 Section 4.2.1.4, the tails of distributions often drive structural failures, so it is important to investigate the confidence in the underlying probability distributional form and whether the specified distribution fits the underlying data well in the tails. Inputs with substantial uncertainty about the probability distribution or uncertainty representation may be candidates for future sensitivity studies to understand the impact of the chosen distribution on analysis results.

For the next sensitivity study, the shapes of select distributions were changed to assess the effect on the results. The inputs permutation routine was invoked to automatically generate new distributions for the following input variables:

- initial_cc_full_length
- initial_cc_depth

To produce the new distributions, the original mean values were preserved, and the standard deviations were doubled. Additionally, the quantiles of the original upper and lower truncation values were used in the new distribution to determine the new upper and lower truncation values.

With these updates, the univariate permutation importance results using ML for the top ten most important input variables are shown in Table 3-10 for the “is_ruptured” QoI and Table 3-11 for the “total_leak_rate” QoI, and the multivariate results are shown in Table 3-12. From these results it is seen that the most important input variable changes from “WRS_axial_premitigation_pt01” to “initial_cc_full_length,” although the former is still ranked as the second-most important input variable. The variable, “initial_cc_full_length,” represents the extent to which the inside surface of the pipe has cracked, and it reasonable that this input would affect the results, for example, because a larger extent of cracking at the start of the simulation would be expected to lead to more ruptures by the end of the simulation. Again, the ML method was able to quantity the effect of changing the inputs. This type of sensitivity study could also be performed in the reverse, that is, by tightening the distribution tails, perhaps to study the effect that some increased understanding of the inputs could help in reducing their relative importance.

Table 3-10 Top ten input permutation importance values for the “is_ruptured” QoI using random forest regression model after changing probability distribution tails

Probabilistic Input	Permutation Importance Value
initial_cc_full_length	0.9480
WRS_axial_premitigation_pt01	0.4888
initial_cc_depth	0.3325
WRS_axial_premitigation_pt26	0.0183
WRS_axial_premitigation_pt12	0.0080
WRS_axial_premitigation_pt24	0.0068
weld_material_PWSCC_growth_component_to_component_variability_ factor_fcomp	0.0057
WRS_axial_premitigation_pt25	0.0057

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt22	0.0052
random	0.0050

Table 3-11 Top ten input permutation importance values for the "total_leak_rate" QoI using random forest regression model after changing probability distribution tails

Probabilistic Input	Permutation Importance Value
initial_cc_full_length	0.6574
WRS_axial_premitigation_pt01	0.3981
initial_cc_depth	0.1755
WRS_axial_premitigation_pt08	0.1550
WRS_axial_premitigation_pt12	0.0389
WRS_axial_premitigation_pt10	0.0317
left_pipe_material_elastic_modulus	0.0312
WRS_axial_premitigation_pt23	0.0286
WRS_axial_premitigation_pt03	0.0283
right_pipe_material_ultimate_strength	0.0262

Table 3-12 Top ten input permutation importance values for all QoIs using random forest regression model after changing probability distribution tails

Probabilistic Input	Permutation Importance Value
initial_cc_full_length	0.8775
WRS_axial_premitigation_pt01	0.4616
initial_cc_depth	0.3111
weld_material_PWSCC_growth_component_to_component_variability_factor_fcomp	0.0207
WRS_axial_premitigation_pt26	0.0201
WRS_axial_premitigation_pt12	0.0142
WRS_axial_premitigation_pt24	0.0133

Probabilistic Input	Permutation Importance Value
WRS_axial_premitigation_pt22	0.0114
WRS_axial_premitigation_pt25	0.0102
WRS_axial_premitigation_pt03	0.0102

3.4 Summary

The RES staff and SNL demonstrated that ML can provide effective support for PFM analyses by automating sensitivity analyses and sensitivity studies. The ML approach used here has the benefit of using existing, off-the-shelf code that is available in several well-vetted Python packages. The approach also easily scales from univariate to multivariate applications. The ML model run-time, including both training and computing the permutation importance values, is longer than for linear regression but comparable. The amount of code developed for xLPRbot was small at less than 1,000 lines of code, and it can be run on a typical laptop without the need for high-performance computing hardware. The approach also facilitates comparing many different ML approaches (i.e., any that have been previously implemented in Python packages) with standard statistical methods. For simple linear problems, the linear regression model is still preferred for its simplicity and speed, but for more complicated non-linear dynamics, ML approaches provide alternatives with potential benefits in speed, code simplicity, and extensibility.

4 CONCLUSIONS

The objective of the RESbot FFR project was to conduct feasibility studies for developing autonomous researchers to assist the RES staff. Use Case 1 explored RESbot development for knowledge mining using tools available from two large commercial vendors. Use Case 2 explored RESbot development for intelligent search of the numerical solution space to support modeling and simulation applications. Through the two uses cases, the RES staff explored aspects of AI, specifically NLP and ML, to automate various research tasks.

The commercial NLP tools show promise for knowledge mining applications; however, the results from this study were inconclusive. The staff observed the advantages of having a high-quality user interface and user experience. Additionally, the staff observed that training the NLP models requires a significant initial investment of the human resources, although this could be reduced somewhat by computer-guided processes and an effective user interface and user experience. It is noted that the staff's explorations were rather limited due to resource constraints. There is accelerated progress in developing AI/ML solutions for NLP in the commercial sector, such that it is likely that the type of bot envisioned for this project may be fully functional within 5 years. As such, other commercial models may be worth exploring in future work.

More promise was shown with development of the open-source tools to support intelligent search for modeling and simulation applications. Here, the RES staff was able to integrate ML with a PFM analysis. The ML results were used to understand the importance of input variables and to automate sensitivity analyses and sensitivity studies. Such applications have immediate use to support research efforts and to review licensing applications that rely on PFM consistent with RG 1.245. The approach could also be extended to other probabilistic modeling and simulation applications.

5 REFERENCES

[1] King, R. D., et al., “Functional Genomic Hypothesis Generation and Experimentation by a Robot Scientist,” *Nature,* vol. 427, pp. 247-252, 2004.

[2] King, R.D., “The Robot Scientist Adam,” *Computer,* vol. 42, pp. 46-54, 2009.

[3] Williams, K., et al., “Cheaper Faster Drug Development Validated by the Repositioning of Drugs Against Neglected Tropical Diseases,” *Journal of the Royal Society Interface,* vol. 12, no. 104, 2015.

[4] Burger, B., et al., “A Mobile Robotic Chemist,” *Nature,* vol. 583, pp. 237-241, 2020.

[5] Gongora, A. E., et al., “A Bayesian Experimental Autonomous Researcher for Mechanical Design,” *Science Advances,* vol. 6, no. 15, 2020.

[6] Jordan, B. and Leong, S., “The Spectrum of Artificial Intelligence,” *Future of Privacy Forum,* 2021.

[7] OpenAI, "OpenAI" [Online]. Available: https://openai.com/. [Accessed December 16, 2022].

[8] NRC, Regulatory Guide 1.245, “Preparing Probabilistic Fracture Mechanics Submittals,” Rev. 0, January 2022, ADAMS Accession No. ML21334A158.

[9] NRC, NUREG-2247, “Extremely Low Probability of Rupture Version 2 Probabilistic Fracture Mechanics Code,” August 2021, ADAMS Accession No. ML21225A736.

[10] NRC, Regulatory Guide 1.99, “Radiation Embrittlement of Reactor Vessel Materials,” Rev. 2, May 1988, ADAMS Accession No. ML003740284.

[11] NRC, 10 CFR Part 50, Domestic Licensing of Production and Utilization Facilities, Washington, DC, 2022.

[12] NRC, TLR-RES/DE/CIB-2020-09, “RG 1.99 Revision 2 Update FAVOR Scoping Study,” October 2020, ADAMS Accession No. ML20300A551.

[13] NRC, TLR-RES/DE/REB-2021-14-R1, “Probabilistic Leak-Before-Break Evaluations of Pressurized-Water Reactor Piping Systems using the Extremely Low Probability of Rupture Code,” April 2022, ADAMS Accession No. ML22088A006.

[14] NRC, NUREG/CR-7278, “Technical Basis for the use of Probabilistic Fracture Mechanics in Regulatory Applications,” January 2022, ADAMS Accession No. ML22014A406.

[15] Antolovich, A. and Antolovich, B., “An Introduction to Fracture Mechanics,” in *ASM Handbook, Volume 19, “Fatigue and Fracture*,” Materials Park, OH, ASM International, 1996, pp. 371-380.

[16] Anderson, T., Fracture Mechanics Fundamentals and Applications, Boca Raton: CRC Press, 2005.

[17] Goodfellow, I., et al., Deep Learning, Cambridge, MA: MIT Press, 2016.

[18] Pedregosa, F., et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research,* no. 12, p. 2825–2830, 2011.

[19] Breiman, L., “Random Forests,” *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[20] Hinton, G. E., et al., “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Computation,* vol. 18, no. 7, p. 1527–1554, 2006.

[21] scikit-learn Developers, “Feature importances with a forest of trees,” [Online]. Available: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html. [Accessed December 12, 2022].

[22] The Matplotlib Development Team, “Matplotlib: Visualization with Python” [Online]. Available: https://matplotlib.org/. [Accessed December 12, 2022].

[23] L. Breiman, Classification and Regression Trees, First Ed., New York: Routledge, 1984.

[24] Altmann, A., et al., “Permutation Importance: A Corrected Feature Importance Measure,” *Bioinformatics,* vol. 26, no. 10, p. 1340–1347, 2010.

[25] Eckert-Gallup, A., et al., SAND2017-2854, “xLPR Scenario Analysis Report,” Sandia National Laboratories, Albuquerque, New Mexico, March 2017, ADAMS Accession No. ML19337B979.

[26] NRC, TLR-RES/DE/CIB-2021-11, “Sensitivity Studies and Analyses Involving the Extremely Low Probability of Rupture Code,” May 14, 2021, ADAMS Accession No. ML21133A485.