Literary Review:

ACM (4) papers:

QUERY:

Abstract:("machine learning") AND AllField:("imperative programming") AND AllField:("functional programming")

Paper	*Machine Learning*	*Imperative/ Functional*	*Topic(s)/ Keyword(s)*
Yadu Babuji et al., “Parsl: Pervasive Parallel Programming in Python” (2019)	*Other*	*Briefly mentions “functional programming”*	*Parallelism, Parallel computing*
Jinlang Wei et al., “Automating Dependence-Aware Parallelization of Machine Learning Training on Distributed Shared Memory” (2019).	Yes	Other	Parallelism, distributed shared memory
Adilla Susungi et al., “Meta-programming for cross-domain tensor optimizations” (2018).	YES	Functional	Tensor operations/optimizations, ML,
Hengchu Zhang, “Fuzzi: a three-level logic for differential privacy” (2019).	No	Other	Differential privacy, programming logic, static analysis

arXiv (14 + 5):

Query 1: functional and imperative programming in machine learning (abstracts)

Paper	Machine Learning	Programming Paradigm	Topic(s)/Keyword(s)
Aaditya Naik et al., “MBD: Interactively Querying Datasets and Models” (2023).	Other	“Uses functional constructs” as part of the implementation of MBD	Debugging, ML
Joao Flach and Luis C. Lamb, “A Neural Lambda Calculus: Neurosymbolic AI meets the foundations of computing and functional programming” (2023).	Yes	Functional	Machine learning, lambda calculus, NNs
Oluwatosin Ogundare et al., “No Code AI: Automatic generation of Function Block Diagrams from documentation and associated heuristic for context-aware ML algorithm training” (2023).	Yes	None	Machine learning, Automatic code-generation
Roland Olsson et al., “Automatic Synthesis of Neurons for Recurrent Neural Nets” (2022).	Yes	Functional	Automatic programming, neuron synthesis
Hoai An Le Thi,	Yes	Other	DCA

“ONLINE STOCHASTIC DCA WITH APPLICATIONS TO PRINCIPAL COMPONENT ANALYSIS” (2021).			
Joshua Clune, “Program Equivalence for Assisted Grading of Functional Programs” (2020).	No	Other	Assisted/ Automated Grading
Kiara Grouwstra, “Type-driven Neural Programming by Example” (2020).	Yes	functional	Program synthesis, PBE, Neural Programming
Andrew K. Lampinen and James L. McClelland, “Zero-shot adaptation by homoiconic meta-mapping" (2019).	Other	“Drew inspiration from functional programming”	Meta-learning, meta-mapping, deep learning
Maria Peifer et al., “Sparse multiresolution representations with adaptive kernels” (2019).	Other	Functional, mathematically that is.	RKHS, multikernel learning
Gregor Ulm et al., “Functional Federated Learning in Erlang (ffl-erl)” (2019).	Yes	Functional... and an indirect comparison between the two paradigms by comparing erlang with C	Machine learning, numerical Computing, functional programming, Distributed computing

Daniel Berenyl et al., “Towards scalable pattern-based optimization for dense linear algebra” (2018).	Other	Functional	Automatic optimizations, linear algebra
Sooraj Bhat et al., “Deriving Probability Density Functions from Probabilistic Functional Programs” (2017).	Other	None	Probabilistic density functions, math
John K. Feser et al., “Differentiable Functional Program Interpreters” (2017)	Other	Functional	PBE (Programming by Example), Programming language synthesis

Query 2:

Started with … papers in total with this query: “Machine Learning” AND “imperative Programming”:

Paper	Machine Learning	Imperative/ functional	Topic(s)/ Keyword(s)
Joao Flach and Luis C. Lamb, “A Neural Lambda Calculus: Neurosymbolic AI meets the foundations of computing and functional programming” (2023).	-	-	-
Griffin Dietz et al., “ARtonomous: Introducing Middle School Students to Reinforcement Learning	Yes	Other	Ml, RL, Learning tools, robotics

Through Virtual Robotics” (2022)			
Stefano Calzavara, “Certifiying Decision Trees Against Evasion Attacks by Program Analysis” (2020).	Yes	**Other**	Evasion attacks, Program analysis, Security (in ML).
Aisha Mohamed, “RDFFrames: Knowledge Graph Access for Machine Learning Tools” (2021).	Yes	both	RDF, knowledge graphs, ML
Kai-Wei Chang, “A Credit Assignment Compiler for Joint Prediction” (2016)	YES	Other	Joint prediction in Ml, credit assignment problem.

Query 3:

Started with … papers in total with this query: “Machine Learning” AND imperative AND declarative:

Paper	Machine Learning	Imperative/ functional	Topic(s)/ Keyword(s)
Michael Innes, “Functional or imperative? On pleasant semantics for differentiable programming languages” (2022).	YES	Both are compared in the paper.	Machine learning, language design.
Tushar Swamy et al., “Homunculus: Auto-Generating Efficient Data-Plane ML Pipelines for	Yes	Functional/ declarative	Network operations, ML.

Datacenter Networks” (2022).			
Yen-Hsiang Chang et al., “MLHarness: A Scalable Benchmarking System for MLCommons” (2022).	Yes	none	Benchmarking, ML
Aisha Mohamed, “RDFFrames: Knowledge Graph Access for Machine Learning Tools” (2021).	-	-	-
Tianqi Chen, “MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems” (2015).	Yes	Blends both paradigms: “it blends **declarative** symbolic expression with **imperative** tensor computation. It offers auto differentiation to derive gradients”.	Heterogeneous Distributed Systems, Ml

IEEM (9 +):

Query (read some of them, trouble with access sometimes):

("Abstract":"Machine Learning") AND ("Abstract":"Functional Programming")
This query from IEEM returns a total of nine papers, five of them in the domain of machine learning and two to three of that paper discuss functional programming and one of them discuss imperative programming (paper 4).

Paper	Machine Learning	Imperative/ functional	Topic(s)/ keyword(s)
Dinesh C. Dobhal et al., “Involvement of Functional Programming in Language Processing and Machine Learning” (2023).	Yes	functional	ML, language processing
Matthew Fenwick et al., “An Open-Source Sandbox for Increasing the Accesibility of Functional Programming to the Bioformatics and Scientific Communities” (2012).	Other	Functional	Learning, programming exercises
Gianluca Aguzzi, “Research directions for Aggregate Computing with Machine Learning” (2021).	Yes	Functional	Aggregate Computing, Machine Learning
Raul Castro Fernandez et al., “Java2SDG: Stateful big data Processing for the masses” (2016).	Yes	Imperative	Stateful data processing, ML
Amin Nazir Nagiwale et al., “Design of self-adjusting algorithm for data-intensive MapReduce applications” (2015)	Yes	Partly functional	Adaptive algorithms, Mapreduce, ML
Oluwatosin Ogundare et al., “No Code AI: Automatic generation of Function Block Diagrams from documentation and associated heuristic for context-aware ML	-	-	-

algorithm training” (2023).			
Miguel Millan et al., “Connecting Cloud Computing and Machine Learning Through Functional Situation-Awareness: A User-Centric Smart Monitoring Application" (2023).	Partly	Functional	Smart monitoring, cloud computing, ML
Richard Roberts et al., “Synthesis of incidental detail as composable components in a functional language” (2013).	Other	Functional	Incidental object details, automatic object generation
Cristian Urlea et al., “Efficient FPGA Cost-Performance Space Exploration using Type-Driven Program Transformations” (2019).	Other	Functional	DSE (Design Space exploration), FPGA (Field Programmable Gate Arrays), functional programming, transformation optimization

Papers that were mentioned in the Bachelor Proposal, and particularly the ones we deem important in helping us answering our research questions:

Paper	Machine	Imperative/	Topic(s)/

	Learning	Functional	Keyword(s)
Miguel Fernandez de Alarcon Gervas et al.,”Functional programming paradigm for machine learning algorithms in data mining. B.S. thesis” (2019).	Yes	Functional	Performance differences between programming paradigms, data mining, ML
Alex Gyori et al., “Crossing the gap from imperative to functional programming through refactoring. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pages 543–553” (2013).	No	Both	Lambda calculus, Code Refraction, Code Quality
Pietro Ehrlich et al., “Functional programming paradigms in reinforcement learning problems” (2022).	Yes	functional	Racket, Code Refraction, ML/RL
Robert Dyer and Jigyasa Chauhan, “An exploratory study on the predominant programming paradigms in python code. In Proceedings of the 30th ACM Joint European Software	No	Both	Data-mining, language features bias

Engineering Conference and Symposium on the Foundations of Software Engineering, pages 684–695,” (2022).			
Dino Alic, Samir Omanovic, and Vaidas Giedrimas, “Comparative analysis of functional and object-oriented programming. In 2016 39th International Convention on Information and Communication Technol-ogy, Electronics and Microelectronics (MIPRO), pages 667–672. IEEE,” (2016).	No	Both	Comparative analysis of programming paradigms, OOP vs FP.
Kim Svensson Sand and Tord Eliasson, “A comparison of functional and object-oriented programming paradigms in Javascript” (2017).	Yes, use of ML algorithms used for the comparison (not all).	Functional and OOP	Javascript, programming paradigms, ML
Per Jernlund and Martin Stenberg, “Functional and imperative object-oriented programming in theory and practice:	No	Both	Online forum analysis, programming paradigms

A study of online discus- sions in the programming community,” (2019).			

Data-Mining:

Top 20 (DATA MINING):

Project/Repository	Language usage	Stars	Forks	Imperative Tokens	Functional Tokens
Tensorflow/tensorflow	C++ (56.2%), Python (27.1%), etc.	179k stars	89.2k	37.2k	6.7k
huggingface/transformers	Python (99.5%)	116K	23.3k	5.6k	2.5k
Josephmisiti/awesome-machine-learning	Python (100%)	61.6k	14.4k	3	0
Microsoft/ML-For-Beginners	HTML (92.7%), Jupyter Notebook (7.2%)	60.8k	12.7k	785	147
fighting41love/funNLP	Python (100%)	59.4K	13.8k	24	4
Scikit-learn/scikit-learn	Python (92.3%), Cython (5.8%), C++ (1.2%), etc.	56.6k	25k	3.1k	677
TheAlgorithms/C-Plus-Plus	C++ (98.6%)	26.8k	6.8k	1k	36
Ageron/handson-ml2	Jupyter	26k	12.3k	67	13

	Notebook (100%)				
Gradio-app/gradio	Python (61.2%), Svelte (20%), TypeScript (15.9%)	24.7k	1.7k	1.6k	237
eriklindernoren/ML-From-Scratch	Python (100%)	22.6k	4.4k	134	36
Trekhleb/homemade-machine-learning	Jupyter Notebook (99.4%)	22k	4k	144	17
PaddlePaddle/Paddle	C++ (50%), Python (41.2%), Cuda (6.6%), etc.	21.1k	5.4k	31.7k	4.5k
RasaHQ/rasa	Python (99.3%)	17.3k	4.5k	1.6k	429
onnx/onnx	Python (55.8%), C++ (42.9%)	16k	3.6k	1.1k	217
aleju/imgaug	Python (100%)	13.9k	2.4k	433	140
mlflow/mlflow	Python (53.4%), Javascript (27.3%), TypeScript (15.6%), etc.	15.9k	3.7k	4.4k	813
Unity-Technologies/ml-agents	C# (54.6%), Python (40.4%), Jupyter Notebook (4.7%), etc.	15.7k	4k	1.6k	324
microsoft/LightGBM	C++ (51%), Python	15.7k	3.8k	706	170

	(20.5%), R (12.2%), etc.				
apache/brpc	C++ (94.4%), Perl (1.4%).	15.5k	3.8k	1.7k	324
Ddbourgin/numpy-ml	Python (100%)	14.3k	3.6k	318	44

Imperative-adjecent repos (python projects), Top 10:

Query-URL: https://github.com/search?q=Machine+Learning+AND+python+language%3APython&type=repositories&s=forks&o=desc&l=Python (we will use a proxy as a placeholder and that will be term, “python,” instead of the term, “imperative” in the query-URL, to make it fair and honest of a comparison as possible. Scala vs python top 10 projects in terms of forks/stars. (how to weigh it properly?)

Repo	Languages?	Forks	Stars	
Scikit-learn/scikit-learn	--	25k	56.7k	
huggingface/transformers	--	23.3k	117k	
fighting41love/funNLP		13.8k	59.8k	
Lazyprogrammer/machine_learning_examples	--	6.2k	7.8k	
Yorko/mlcourse.ai	--	5.7k	9.2k	
Jack-Cherish/Machine-Learning	--	5k	8.1k	
gunthercox/ChatterBot	--	4.4k	13.7k	
Lawlite19/MachineLearning_Python	--	2.4k	6.2k	
microsoft/qlib	--	2.3k	13k	
microsoft/nni	--	1.8k	13.4k	

Functional-adjecent repos (Scala), Top 10:

Query-URL: *https://github.com/search?q=Machine+learning+AND+scala+language%3AScala&type=repositories&s=forks&o=desc&l=Scala*

Repo	Languages?	Forks	Stars	
apache/predictionio	--	2k	12.6k	
microsoft/SynapseML	--	824	4.9k	
salesforce/TransmogrifAI		397	2.2k	
prnicolas/ScalaMl	--	86	151	
Ymetarank/metarank	--	75	1.9k	
spotify/featran	--	72	466	
PacktPublishing/Scala-Machine-Learning-Projects	--	56	45	
Adornes/spark_scala_ml_examples	--	52	78	
Tailhq/DynaML	--	51	198	
spider-123-eng/Spark	--	42	55	

Experiment:

(Linear Regression):

Languages	Execution Time (in ms)	Memory Usage (in MB)	Memory usage when forcing the garbage collector (this, strictly, concerns the Scala code only) (we used a profiler for the memory in python).	Lines of Code (spaces and libraries included)
Python	Avg of 5: (108, 76, 77, 75, 77) = 76 ms	Avg of 5: (0.3, 0.5, 0.0, 0.1, 0.5)/5 =	--	68 LOC

		0.28 MB		
Scala	Avg of 3 (+/- 0.001 ms): 100 ms, 100ms, 100ms	Avg of 3: 197 MB, 185 MB, 185 MB = 189 MB	Avg of 3: 1 MB, 1MB, 1MB	102 LOC + benchmark file (72)

(Logistic Regression (classification)):

Languages	Execution Time (in ms)	Memory Usage (in MB)	Memory usage when forcing the garbage collector (this, strictly, concerns the Scala code only) (we used a profiler for the memory in python).	Lines Of Code (spaces, libraries included)
Python	Avg of 5:(72, 73, 75, 75, 73) = 73.6 ms	Avg of 5: (0.4 + 0.3 + 0.1 + 0.2 + 0.1) /3 = 0.36 MB	--	53 LOC
Scala	Avg of 3 (+/- 0.001):) = 333.33 ms	Avg of 3: (149, 160, 147)/3= 152 MB	Avg of 3: (6, 10, 11) / 3 = 9	79 LOC + benchmark file (81)