PROJECT IDEAS · M.TECH · MACHINE LEARNING · 2026-27

M.Tech projects in machine learning, *with base papers*

These 13 M.Tech projects in machine learning are each built on an open-access IEEE paper from 2023 to 2026, checked against its DOI record. Every card names the data and tools, and **one change to measure for your thesis, with the comparison table it produces**. Pick one, reproduce the paper, then extend it.

Last updated 2026-11-12

We matched each paper to its Crossref DOI record on 12 November 2026; titles, journals and years appear exactly as IEEE registered them, and each card says only what the paper's abstract says. For base papers in more domains, see the 28 IEEE base papers by domain; for M.Tech topics beyond machine learning, see our M.Tech project ideas.

KEY POINTS

- **13 M.Tech machine learning ideas,** each built on an open-access IEEE paper from 2023 to 2026. Thirteen appeared in IEEE Access; the lung nodule paper appeared in the IEEE Open Journal of Engineering in Medicine and Biology.
- **Four groups:** tabular data and time series (4), health data (3), NLP and LLMs (4), and trustworthy and efficient ML (4).
- **Every card gives one M.Tech extension** you can measure, and the comparison table it puts in your thesis.
- **Reproduce first.** Your extension is judged against the result you reproduced, on the same split and metric.
- **We quote no accuracy figures.** Read each paper's own results and match them yourself before you claim an improvement.

What makes a good M.Tech machine learning project?

Three things: a base paper you can rebuild, one change you can measure, and results someone else could repeat.

- **A base paper you can rebuild.** Recent, open access, on data you can download, and clear about its split, settings and metric. Code from the authors helps a lot. Our base paper selection guide walks through every check.
- **One measured extension.** A single change, such as a new model, a second dataset, a robustness test, calibrated uncertainty or a lighter model, tested on the same split as the original. One change measured well is worth more than five changes nobody can separate.
- **Reproducibility.** Fixed random seeds, the exact dataset version, library versions written down and every run logged. Your methodology chapter is written from that log.

This is where M.Tech differs from B.Tech. A B.Tech project can end with a working demo of a known method. An M.Tech thesis has to show, in a results table, that your change did something the base paper did not.

Which machine learning projects suit M.Tech in 2026-27?

Thirteen ideas in four groups. Each card states the problem and what you build and measure. Open “The base paper and the M.Tech extension” to see the paper, linked by its DOI, then the data, the tools and one change to measure, with the table it produces.

Topic ideas, not delivered projects

Each card is built on an open-access IEEE paper, checked against its DOI record on Crossref, and describes only what the paper's abstract states. The project around it is an idea to discuss with your guide. Our **8 delivered case studies** are M.Tech and M.E. projects, each shown with its real paper pages, screens and results.

Tabular data and time series

Most of these run on a laptop CPU. When the data is ordered in time, split it by time and never at random, or the model is tested on a past it has already seen the future of.

TOPIC IDEA · M.TECH · TABULAR AND TIME SERIES

Gradient boosting or deep learning on tabular data: XGBoost, TabNet and TabR

Problem: Most records in Indian banks, hospitals and government offices are tables, and it is still unsettled when a deep model is worth its extra cost over gradient-boosted trees.

Output: XGBoost, TabNet and the retrieval-based TabR trained on the same benchmark datasets and compared on accuracy and AUC for classification and on MAE, RMSE and R^2 for regression.

− The base paper and the M.Tech extension

Base paper: Tabular Data Classification and Regression : XGBoost or Deep Learning with Retrieval-Augmented Generation, J. Pasaribu et al., IEEE Access, 2024
Data, tools and M.Tech extension: Data and tools: the benchmark datasets listed in the paper's experiments section; Python with XGBoost, the pytorch-tabnet package and the public TabR code from its original authors. M.Tech extension: give all three models the same tuning budget with Optuna and record training time, to test whether TabR's lead in classification survives fair tuning; the thesis table shows every metric and the training time for each model, under the paper's settings and under equal tuning.

TOPIC IDEA · M.TECH · TABULAR AND TIME SERIES

Probabilistic household load forecasting with a quantile LSTM

Problem: Indian distribution companies are fitting smart meters in homes, and a single-number load forecast hides how uncertain each household's demand really is.

Output: Households grouped by load pattern, then a quantile LSTM with a log-cosh quantile loss that gives a day-ahead load interval at 15-minute and 1-hour steps, scored on prediction interval coverage against quantile regression forest, QRNN and quantile gradient boosting.

− The base paper and the M.Tech extension

Base paper: Enhancing Short-Term Electric Load Forecasting for Households Using Quantile LSTM and Clustering-Based Probabilistic Approach, Z. Masood et al., IEEE Access, 2024
Data, tools and M.Tech extension: Data and tools: the abstract does not name the household dataset, so check the paper, or use a public multi-household smart-meter set such as the London smart meter data on Kaggle and say so; Python with PyTorch, scikit-learn and tslearn for the clustering. M.Tech extension: add a conformal calibration step on top of the quantile LSTM so the intervals reach their stated coverage; the thesis table gives coverage (PICP) and average interval width for every model, before and after calibration.

TOPIC IDEA · M.TECH · TABULAR AND TIME SERIES

Aircraft engine remaining useful life with a lighter Transformer fusion model

Problem: Maintenance planners want to know how many cycles an engine has left, and fixed service intervals ignore how each engine is actually wearing.

Output: A Transformer with CNN layers in its encoder and LSTM layers in its decoder that predicts the remaining cycles of each engine from its sensor history, scored on RMSE and the C-MAPSS score against plain LSTM and CNN models.

− The base paper and the M.Tech extension

Base paper: Novel Transformer-based Fusion Models for Aero-engine Remaining Useful Life Estimation, Q. Hu et al., IEEE Access, 2023
Data, tools and M.Tech extension: Data and tools: NASA's C-MAPSS turbofan simulation data, in four subsets from FD001 to FD004; Python with PyTorch. M.Tech extension: prune the fusion model, or replace its full attention with a lighter one, to halve its parameters and measure what accuracy it gives up; the thesis table lists RMSE, score, parameter count and inference time on all four subsets for an LSTM baseline, the paper's model and your lighter version.

Deep learning for health data

Keep each patient’s records in one split only. A model that has seen other heartbeats or EEG windows from the same patient in training will look better than it is.

TOPIC IDEA · M.TECH · HEALTH DATA

Heartbeat arrhythmia classification that holds up in noisy ECG

Problem: Rural clinics in India can record an ECG but often have no cardiologist on hand, so a screening model must run on cheap hardware and still catch rare abnormal beats.

Output: A two-layer 1D CNN with squeeze-and-excitation attention that labels each heartbeat from per-beat features, with SMOTE on the training data only, reported as per-class precision and recall, the parameter count and the INT8 model size.

The base paper and the M.Tech extension

Base paper: A Lightweight Channel-Attention CNN for Robust Beat-Level Arrhythmia Detection, M. S. Tahsin et al., IEEE Access, 2026
Data, tools and M.Tech extension: Data and tools: the four PhysioNet databases the paper tests on, which are MIT-BIH Arrhythmia, MIT-BIH Supraventricular Arrhythmia, INCART-2 and the Sudden Cardiac Death Holter database; Python with the wfdb package, and TensorFlow Lite or PyTorch for the quantisation. M.Tech extension: add electrode-motion and muscle noise from PhysioNet's MIT-BIH Noise Stress Test Database at fixed signal-to-noise ratios, then train with noise augmentation; the thesis table gives macro F1 for the paper's model and your noise-trained model on clean records and at each noise level.

TOPIC IDEA · M.TECH · HEALTH DATA

Epileptic seizure detection from raw EEG with fewer false alarms

Problem: Reading a long EEG recording for seizures by eye takes a neurologist hours, and a detector that raises frequent false alarms adds to the work instead of cutting it.

Output: A CNN that takes raw EEG straight in and looks at it at several scales through inception and residual blocks, with no feature extraction step, tested per subject and across subjects on the Bonn and CHB-MIT datasets, with F1-score as the main measure.

The base paper and the M.Tech extension

Base paper: Less Parameterization Inception-Based End to End CNN Model for EEG Seizure Detection, K.-K. Shyu et al., IEEE Access, 2023
Data, tools and M.Tech extension: Data and tools: the Bonn EEG recordings, which are public, and CHB-MIT, the children's scalp EEG database on PhysioNet; MNE reads its EDF files into Python, and PyTorch trains the model. M.Tech extension: add post-processing that raises an alarm only after several consecutive seizure windows, and report the measures a clinician asks for; the thesis table shows sensitivity, false alarms per hour and detection delay for each CHB-MIT patient, with and without the post-processing.

TOPIC IDEA · M.TECH · HEALTH DATA

Explainable diabetes risk prediction: comparing LIME and SHAP

Problem: Diabetes is common in India, and a doctor will not act on a risk score unless the model can show which of the patient's answers raised it.

Output: Logistic regression and random forest models that predict diabetes from health survey answers, with LIME and SHAP explanations for single patients and for the whole model, and a comparison of where the two explainers agree and differ.

− The base paper and the M.Tech extension

Base paper: A Comparative Analysis of LIME and SHAP Interpreters With Explainable ML-Based Diabetes Predictions, S. Ahmed et al., IEEE Access, 2025

Data, tools and M.Tech extension: Data and tools: the 253,680 survey responses the abstract describes, with the source named in the paper; Python with scikit-learn and the lime and shap packages. M.Tech extension: measure how stable the explanations are by rerunning LIME and SHAP with different seeds and background samples, and add a gradient-boosted model; the thesis table gives accuracy and recall for each model beside how often its top five features stay the same across runs and across the two explainers.

NLP and large language models

The classical NLP ideas run on a CPU; the LLM ideas need a GPU. State the model size and the GPU you used in the thesis, because examiners ask.

TOPIC IDEA · M.TECH · NLP AND LLMS

Sentiment analysis of Hindi movie reviews: a stacked ensemble against a transformer

Problem: Indian film fans write many reviews in Hindi, yet most sentiment tools are built for English and Hindi has few labelled datasets.

Output: Tuned classical classifiers on TF-ISF n-gram features, combined in a stacked ensemble that labels a Hindi review positive or negative, compared with each classifier alone on three review datasets.

The base paper and the M.Tech extension

Base paper: Toward Machine Learning Based Binary Sentiment Classification of Movie Reviews for Resource Restraint Language (RRL) —Hindi, A. Sharma and U. Ghose, IEEE Access, 2023
Data, tools and M.Tech extension: Data and tools: the authors' Hindi Language Movie Review set of 10,000 reviews (ask whether they share it) and the public IIT Patna movie and product review sets; Python with scikit-learn and the Indic NLP Library. M.Tech extension: fine-tune a multilingual transformer such as MuRIL or IndicBERT on the same splits; the thesis table compares accuracy, precision, recall, F1 and training time for the stacked ensemble and the transformer on each dataset.

TOPIC IDEA · M.TECH · NLP AND LLMS

Detecting AI-written text with LoRA-tuned open LLMs, in English and Hindi

Problem: Colleges in India now receive assignments written with chatbots, and zero-shot detectors give unstable results.

Output: Open LLMs fine-tuned with LoRA, QLoRA and PLoRA to label a text as human- or machine-written, compared with the same models used zero-shot on accuracy, memory use and training time.

The base paper and the M.Tech extension

Base paper: Enhancing Text-Based Deepfake Detection: A Comprehensive Evaluation of LoRA Fine-Tuned Large Language Models, A. Mohammed and R. Kora, IEEE Access, 2026
Data, tools and M.Tech extension: Data and tools: the four multilingual benchmarks named in the paper; Python with Hugging Face Transformers and PEFT, using small open models with QLoRA on a free Colab or Kaggle GPU. M.Tech extension: build a Hindi and Hinglish test set from human-written text and from text you generate with two LLMs, and measure how a detector tuned on the benchmarks holds up; the thesis table gives accuracy and F1 for zero-shot, LoRA and QLoRA on the original test sets and on your Hindi and Hinglish sets.

TOPIC IDEA · M.TECH · NLP AND LLMS

RAG, fine-tuning or both: how much the retriever decides the answer

Problem: Fine-tuning an LLM on a university's rule book costs GPU time, retrieval adds a step to every answer, and few projects measure which one actually answers better.

Output: Retrieval-augmented generation, fine-tuning and a hybrid of the two, built on one open model and compared on SQuAD, MS MARCO and SQL CREATE TABLE questions for answer quality and compute cost.

− The base paper and the M.Tech extension

Base paper: Unveiling the Power of Large Language Models: A Comparative Study of Retrieval-Augmented Generation, Fine-Tuning, and Their Synergistic Fusion for Enhanced Performance, G. Budakoglu and H. Emekci, IEEE Access, 2025

Data, tools and M.Tech extension: Data and tools: the public SQuAD and MS MARCO question sets, plus the SQL CREATE TABLE statements the authors also tested on; Python with Hugging Face Transformers, PEFT for the fine-tuning and FAISS for the vector index. M.Tech extension: swap the retriever between BM25, dense embeddings and a hybrid of both, and trace how retrieval quality changes the final answer; the thesis table gives the retriever's recall at 5 beside exact match, F1 and time per question for each RAG variant, fine-tuning and

TOPIC IDEA · M.TECH · NLP AND LLMS

Moderating code-mixed Dravidian comments: does transliteration help?

Problem: Comments under Indian videos and posts often mix Tamil, Malayalam or Kannada with English in Roman script, and moderation tools built for English miss the abuse in them.

Output: Embeddings from seven multilingual models, including MuRIL, LaBSE and IndicBERT, fed to classical classifiers with higher weights for the rare classes, compared on six code-mixed datasets.

− The base paper and the M.Tech extension

Base paper: Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages Using Cost-Sensitive Learning Approach, K. Sreelakshmi et al., IEEE Access, 2024

Data, tools and M.Tech extension: Data and tools: the six public sets the paper uses, from DravidianLangTech 2021, HASOC 2020 and HASOC 2021, plus the authors' new Malayalam-English test set; Python with Hugging Face Transformers and scikit-learn. M.Tech extension: transliterate the Roman-script comments into Tamil, Malayalam or Kannada script with an open tool such as IndicXlit before embedding, and measure the effect; the thesis table gives macro F1 for each dataset on the original text and on the transliterated text.

each RAG variant, fine-tuning and the hybrid.

Trustworthy and efficient ML

These ideas ask how far a model can be trusted and how cheaply it runs. That makes the extension easy to measure: coverage, robustness, trainable parameters, memory or time per prediction.

TOPIC IDEA · M.TECH · TRUST AND EFFICIENCY

HER2 breast cancer scoring that says when it is unsure, with conformal prediction

Problem: HER2 status decides whether a breast cancer patient is offered targeted therapy, and a classifier that gives a confident wrong answer on an equivocal slide does more harm than one that asks for a confirmatory test.

Output: Colour, local binary pattern and Haralick features with tree-based classifiers, wrapped in conformal prediction so that uncertain images get a set of labels and are referred for testing, measured by accuracy on the decided cases and the share referred.

The base paper and the M.Tech extension

Base paper: Conformal Prediction for Uncertainty Quantification and Reliable HER2 Status Classification in Breast Cancer IHC Images, S. Pintawong et al., IEEE Access, 2025
Data, tools and M.Tech extension: Data and tools: the paper's 2,570 IHC images from 94 cases come from one pathology institute in Bangkok, so check whether they are shared, and the authors' code is on GitHub; Python with scikit-learn, XGBoost and the MAPIE conformal prediction library. M.Tech extension: replace the handcrafted features with features from a pre-trained CNN and keep the conformal step the same; the thesis table gives

TOPIC IDEA · M.TECH · TRUST AND EFFICIENCY

How easily medical image classifiers are fooled, and what one defence costs

Problem: Hospitals are starting to use deep learning to read scans, and a small change to an image that no doctor would notice can flip the model's answer.

Output: FGSM and PGD attacks applied to whole images and to patches of different sizes, run against four deep learning models on three medical image datasets, with accuracy recorded at each attack strength.

The base paper and the M.Tech extension

Base paper: The Impact of Simultaneous Adversarial Attacks on Robustness of Medical Image Analysis, S. Pal et al., IEEE Access, 2024
Data, tools and M.Tech extension: Data and tools: the three medical datasets and four models named in the paper's experiments section; Python with PyTorch and an attack library such as torchattacks or the Adversarial Robustness Toolbox. M.Tech extension: add adversarial training with PGD examples to one model and measure what it costs in clean accuracy and training time; the thesis table gives clean accuracy and accuracy under FGSM and PGD at each attack strength, before and after the defence.

accuracy on the decided cases and the referral rate at two significance levels for both feature sets.

TOPIC IDEA · M.TECH · TRUST AND EFFICIENCY

Lung nodule classification with LoRA: finding the smallest adapter that works

Problem: Fully fine-tuning a large pre-trained vision model on lung CT nodules needs more GPU memory and labelled scans than most M.Tech projects have.

Output: A large vision model pre-trained with self-supervised learning, adapted with LoRA to label nodules benign or malignant, and compared with full fine-tuning on ROC AUC, trainable parameters and training time.

− The base paper and the M.Tech extension

Base paper: Low-Rank Adaptation of Pre-Trained Large Vision Models for Improved Lung Nodule Malignancy Classification, B. P. Veasey and A. A. Amini, IEEE Open Journal of Engineering in Medicine and Biology, 2025
Data, tools and M.Tech extension: Data and tools: LIDC, which is public on The Cancer Imaging Archive, and NLSTx, whose annotations the authors released on GitHub; Python with PyTorch and the PEFT library. M.Tech extension: vary the LoRA rank and the layers that get adapters, and find the smallest setting that keeps the result; the thesis table gives ROC AUC, trainable parameters, GPU memory and training time for full fine-tuning and for LoRA at three ranks.

Prefer to start from a dataset rather than a paper? See our projects with Kaggle datasets, then look for a recent IEEE paper on the same data.

Choosing an M.Tech ML topic?

Send the idea or base paper, your specialisation and your Stage I and Stage II dates. The consultation is free.

Message in the next 7 h 19 min for a reply today.

Free consultation →

WhatsApp

How does an M.Tech machine learning project run, stage by stage?

Most universities split the M.Tech project into two stages across the second year. The names differ, such as Phase I and Phase II or Dissertation Part I and Part II, but the work falls the same way.

Stage I: literature and reproduction

1

Choose the base paper

and confirm that its data is available and the comp

ute
fits
your
time.

2 **Read around it.**

Colle
ct the
paper
s it
builds
on
and
the
ones
that
cite it,
and
note
what
each
one
chan
ged.
This
beco
mes
your
literat
ure
surve
y.

3 **Reproduce the main result.**

Matc [how to implement a base paper](#)
h the

dataset, split, preprocessing and metric, and put your number beside the paper's. Our guide on covers each step and what to do when the numbers differ.

4 Propose the extension.

Name the one

change the metric and the table you will fill in Stage II.

5

Present Stage I.

The review panel wants the paper, your reproduced result and the plan. The base paper presentation guide shows the slides to make.

Stage II: extension, results, thesis and paper

1 Build the extension

and run it on the same split as your reproducti on

2 Fill the comparison table:

the paper's numb er, your reproducti on and your exten sion on the same metri c, with more

than one random seed where training is unstable.

3 **Write the thesis.**

The results chapter is built around that table. Our guide gives the chapter order and the formatting.

M.Tech thesis format

4 **Write a paper**

if your university or guide asks for one, usually in the IEEE two-column layout. Our [IEEE paper format guide](#) lists the rules.

5 Prepare for the viva.

Be ready to explain every number in your table, and why your change

helpe
d or
did
not.

Which datasets and how much compute do you need?

Every idea here uses public data or data the authors describe in their paper. Where a card says to check the paper, do that before you commit, because the project cannot start without its data.

- **Where public datasets live.** PhysioNet for ECG and EEG, The Cancer Imaging Archive for CT scans, Kaggle and the UCI Machine Learning Repository for tabular data, and Hugging Face Datasets for text. Read each dataset's licence and cite it.
- **Free GPUs.** Google Colab and Kaggle notebooks give free GPU time, with limits on session length and weekly use that change from time to time. Save checkpoints to Drive or a Kaggle dataset, so a dropped session does not cost you a day's training.
- **LLM projects.** Use small open models, load them in 4-bit with QLoRA, and state in the thesis which model size and GPU you used.
- **When you need more.** 3D CT work, large ensembles and long fine-tuning runs are easier on a college GPU server or a paid cloud GPU rented for a few days.

The 13 M.Tech machine learning ideas, with the data each uses and the compute it needs

IDEA	DATA	COMPUTE
XGBoost, TabNet and TabR	Benchmark tabular sets (see the paper)	Laptop CPU for XGBoost; free GPU for TabNet and TabR
Household load forecasting	Household smart-meter data	Laptop CPU or free GPU
Engine remaining useful life	NASA C-MAPSS	Free GPU

IDEA	DATA	COMPUTE
ECG arrhythmia in noise	Four PhysioNet ECG databases	Laptop CPU
EEG seizure detection	Bonn, CHB-MIT	Free GPU
Diabetes LIME and SHAP	253,680 survey responses	Laptop CPU
Hindi review sentiment	HLMR, IIT Patna reviews	Laptop CPU; free GPU for the transformer
AI-written text detection	Four multilingual benchmarks (see the paper)	Free GPU with small models and QLoRA
RAG or fine-tuning	SQuAD, MS MARCO, SQL CREATE TABLE	Free GPU with small models
Code-mixed offensive comments	DravidianLangTech, HASOC	Free GPU for the embeddings; CPU for the classifiers
HER2 conformal prediction	2,570 IHC images (check access)	Laptop CPU
Medical image attacks	Three medical image sets (see the paper)	Free GPU
Lung nodule LoRA	LIDC, NLSTx	College or paid GPU

Compute is our estimate for an M.Tech-sized run, not a figure from the papers. Check each dataset's licence and access terms before you commit.

How can we help with your M.Tech machine learning project?

We help M.Tech students in CSE, IT and AI from Stage I to the final submission. We check the base paper and its data first, reproduce the main result, then design one extension, run it on the same split and write it up. You get the code and results, the thesis in your university's format, an IEEE-format paper if your guide wants one, review presentations and viva preparation, delivered stage by stage to match your reviews.

Start with our M.Tech projects for CSE and AI page. It sets out the scope, the two stages and what you receive. The consultation is free, and you get a written plan and a fixed quote.

More M.Tech domains, each built on verified IEEE base papers: M.Tech projects in image processing, M.Tech projects in embedded systems and IoT, M.Tech

projects in cyber security. For the full service, see our M.Tech projects for CSE and AI.

Frequently asked questions

Which machine learning topic is best for an M.Tech project?

The best topic has a recent open-access base paper, public data, compute that fits your time and one change you can measure on the same test set. Pick an area you and your guide both know, then choose from the cards above. For topics outside machine learning, see our M.Tech project ideas.

Can I do an M.Tech machine learning project without a GPU?

Yes, for several ideas here. The tabular, load forecasting, ECG, diabetes, Hindi sentiment and HER2 ideas run on a laptop CPU. The deep learning and LLM ideas need a GPU, and free Colab or Kaggle notebooks are enough for small models if you save checkpoints often. Our guide to implementing a base paper covers working with limited compute.

How recent should an M.Tech base paper be?

Aim for the last three years, so 2023 or later for a 2026–27 project, and prefer an open-access journal such as IEEE Access so everyone on your review panel can read it without a subscription. Check that its data is public before you commit. Our guide to choosing a base paper covers the rest.

What counts as an M.Tech extension in machine learning?

One change measured on the same split and metric as the base paper: a new model, a second dataset, a robustness test, calibrated uncertainty or a lighter model. The result is a results table that sets the paper's figure, your reproduced figure and your extension's figure in adjacent columns. The M.Tech thesis format shows where that table goes.

Can my M.Tech machine learning project become a research paper?

Yes, if the extension is measured well. A paper needs the same comparison table as your thesis, written up in a shorter form, usually in the IEEE two-column layout. Our research paper writing service covers the writing and a journal shortlist.

Can The Ultimate Project World help with my M.Tech machine learning project?

Yes. Send the idea or base paper, your specialisation and your Stage I and Stage II dates in the free consultation. We reproduce the paper, build and measure the extension, and prepare the thesis, an IEEE-format paper if you need one, and viva preparation. See our M.Tech projects service.

RELATED SERVICES

From a topic to a project you can present

M.Tech projects
A base paper, one measured extension, a paper and the thesis. →

IEEE projects
Start from a published base paper and extend it. →

Research papers
A paper in IEEE, Springer or journal format, and a journal shortlist. →