UNIT-4

Machine Learning and Expert Systems

Practical ML examples, Multisession inductive programming, Expert systems: The success story, Expert systems as AI software, Engineering Expert systems, The lessons of expert systems for engineering AI software.

ML and expert systems are two distinct but interconnected fields within AI.

ML:

ML is a subset of AI that focuses on developing algms and models that enable computers to learn from data and make predictions or decisions without explicit programming.

ML algms use statistical techniques to allow computers to improve their performance on a specific task over time.

SL: The algm is trained on a labeled dataset, where the i/p data is paired with corresponding o/p labels.

The model learns to map i/ps to o/ps based on the provided examples.

USL: The algm is given unlabeled data and must find patterns or relationships within the data on its own.

Clustering and dimensionality reduction are common tasks in USL.

RL:

The algm learns by interacting with an environment and receiving feedback in the form of rewards or penalties.

It aims to find the optimal actions to maximize cumulative reward over time.

Semi SL and self SL:

These are hybrid approaches that leverage both labeled and unlabeled data or generate labels from the data itself.

ML is applied in various domains including image and speech recognition, NLP, recommendation systems, and more.

Expert Systems:

Expert systems are a type of AI that emulates the decision-making ability of a human expert in a specific domain.

They use a KB of human expertise and an inference engine to draw conclusions or make decisions.

Expert systems are rule-based and rely on predefined rules and logical reasoning.

KB: Contains information and rules provided by human experts in a specific domain.

Inference Engine: Responsible for applying the rules and reasoning through the KB to make decisions or draw conclusions.

UI: Facilitates the communication b/w the user and the expert system.

Expert systems are particularly useful in areas where human expertise is critical, such as medical diagnosis, financial analysis, and trouble-shooting complex systems.

ML techniques can be used to automatically learn patterns from data, while expert systems can provide interpretability and explainability in decision-making processes.

Practical ML Examples:-

Inductive generalization techniques have been used to generate expert systems from a set of examples of the expertise.

The current computer induction techniques can already offer a viable knowledge acquisition method if the problem domain is sufficiently simple and well-defined.

Image classification:

Classifying images into categories.

Image recognition in social media, security surveillance, (or) medical imaging.

NLP:

Understanding ~~the~~ and processing human language.

Sentiment analysis and customer reviews.

Chatbots, virtual assistants, social media monitoring.

Speech Recognition:

Converting spoken language into text.

Voice commands for smart devices.

Recommendation Systems:

Providing personalized recommendations.

Streaming services, e-commerce platforms.

Fraud Detection:

Identifying unusual patterns to detect fraudulent activities.

Banking, online transactions.

Healthcare Diagnosis:

Assisting in medical diagnosis based on patient data.

Disease prediction, personalized medicine.

Autonomous Vehicles:

Enabling vehicles to navigate without human intervention.

Object detection, self-driving cars, drones.

Financial Forecasting:

Predicting stock prices & market trends.

Stock trading, investment strategies.

It's important to note that the success of a ML appⁿ often depends on the quality and relevance of the data used for training the models.

Multiversion Inductive Programming :-

A generalized practical s/w engg technique that is emerging from these inductive AI techniques.

ie, PDP networks and decision trees.

These inductive techniques have been viewed as implementing AI techniques that have been demonstrated on a few unrelated and specialized problems.

Many A.I. problems relate to very complex systems, the human body & human language are just two examples. These are termed data-defined problems.

S/w construction is an automatic, algorithmically determined procedure, once the data has been analyzed and organized.

When the initial data processing has been completed, the cost of S/w dev is cheap because machine time for automatic induction is much cheaper than person time for manual algm dev.

The automatic process is also much more controllable, potentially engineerable - the initial conditions determine the outcome.

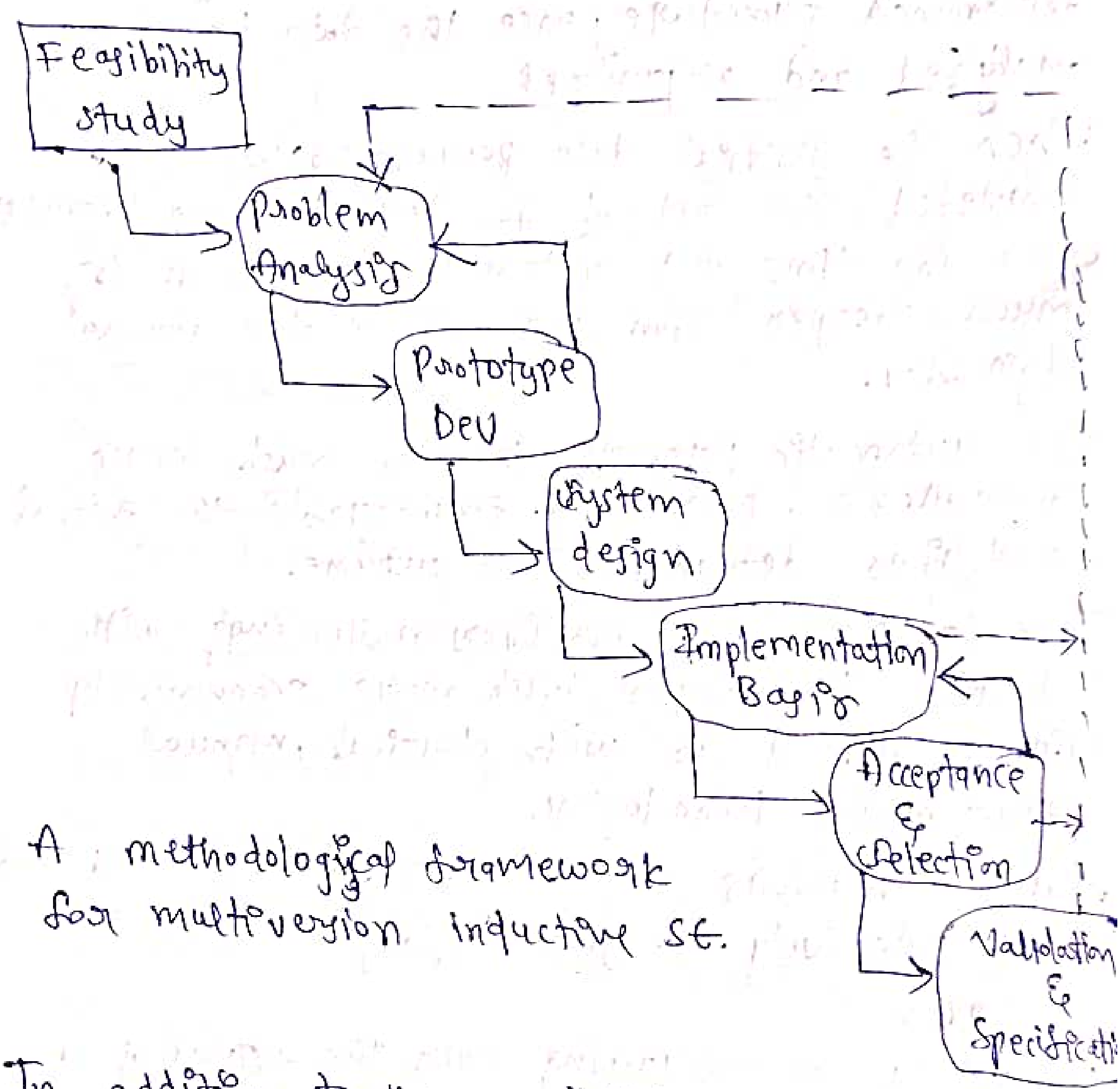
These factors make multiversion S/w engg with inductive technologies much more economically viable than it is with classical, manual programming technologies.

Multiple versions can be routinely constructed and used collectively to yield highly reliable S/w systems.

By voting or averaging over the results of a diverse version set, high performance S/w systems can be built from not very reliable components.

The majority-vote performance of the three-version is better than any of the individual versions because the versions are diverse..

With respect to voting, the necessary diversity is lack of coincident failure.
i.e, no two versions failure on the same i/p.
and lack of identical failure.
i.e, no two versions fail with the same wrong answer.



A methodological framework for multiversion inductive SE.

In addition to the possibility of increasing s/w power through the use of ML, there is the potential use of ML techniques in a new approach to s/w dev methodology: build a basic system and then train it to the required level.

Expert Systems : The Success Story :-

Expert systems had a significant impact on various industries, particularly during the 1970s and 1980s.

They were among the earliest applis of AI and played a crucial role in automating decision-making process in specific domains.

Dendral : Chemistry.

Dendral was one of the first expert systems, developed at Stanford University in the 1960s.

It was designed to analyze mass spectrometry data and identify the molecular structure of organic compound.

MYCIN : Medicine.

Developed in the 1970s at Stanford University, MYCIN was an expert system designed for diagnosing bacterial infections and recommending antibiotic treatments.

PROSPECTOR: Geology and Mineral Exploration.

PROSPECTOR, developed at the Stanford Research institute in the 1970s, was used for mineral exploration.

XCON : Business and Configuration. (DEC)

XCON was developed by Digital Equipment Corporation in the 1980s.

It was used for configuring computer systems based on customer requirements.

While these early expert systems had notable successes, they also faced challenges such as knowledge acquisition bottlenecks, brittleness, and limitations in handling uncertainty.

Expert systems as AI s/w:

It is clear that some expert systems are examples of practical AI s/w, and equally clearly some are not.

The KBs that are known to contain all of the relevant information, that are operated on with a straightforward mechanism of logical inference, and that generate results, which are decidably correct or incorrect on the basis of an abstract decision procedure.

The non-AI expert systems are conventional SE problems implemented by means of AI techniques, which may or may not, be the most effective implementation strategy.

Expert systems are a type of AI s/w designed to emulate the decision-making abilities of a human expert in a specific domain.

These systems are built on a KB consisting of facts, rules, and heuristics provided by human experts, along with an inference engine that applies logical reasoning to draw conclusions or make decisions.

KB:

KB is a central component of an expert system. It contains a repository of domain-specific information, rules, and

heuristics.
This information is typically gathered from human experts in the field.

Inference Engine:

The inference engine is responsible for processing information from the KB.
It uses logical reasoning and inference rules to draw conclusions or make decisions based on the input data.

UI:

UI allows interaction b/w the expert system and the end user.
Users can i/p queries and the system provides responses or recommendations based on its KB and inference engine.
The interface may vary from a CLI to a GUI, depending on the appn.

Knowledge ~~to~~ Acquisition System:

Developing the KB is often on of the most challenging aspects of creating an expert system.
Knowledge acquisition tools or systems assist in gathering, organizing, and inputting the expertise of human domain experts into the KB.

Applications:

Expert systems have been applied in diverse fields including healthcare, finance, engineering, and troubleshooting.
Expert systems continue to play a role in specific applications where explicit knowledge representation and rule-based reasoning are essential.

Engineering Expert systems :-

Some expert systems are definitely AI s/w, and some lesser portions of these are robust and reliable AI s/w.

- Conceptualization — determining problem characteristics.
- Identification — Finding concepts to represent knowledge.
- Formalization — Designing structures to organize knowledge.
- Implementation — Formulating rules that embody knowledge.
- Testing. — Validating rules that embody knowledge.

Identification is determining the scope of the problem to be addressed.

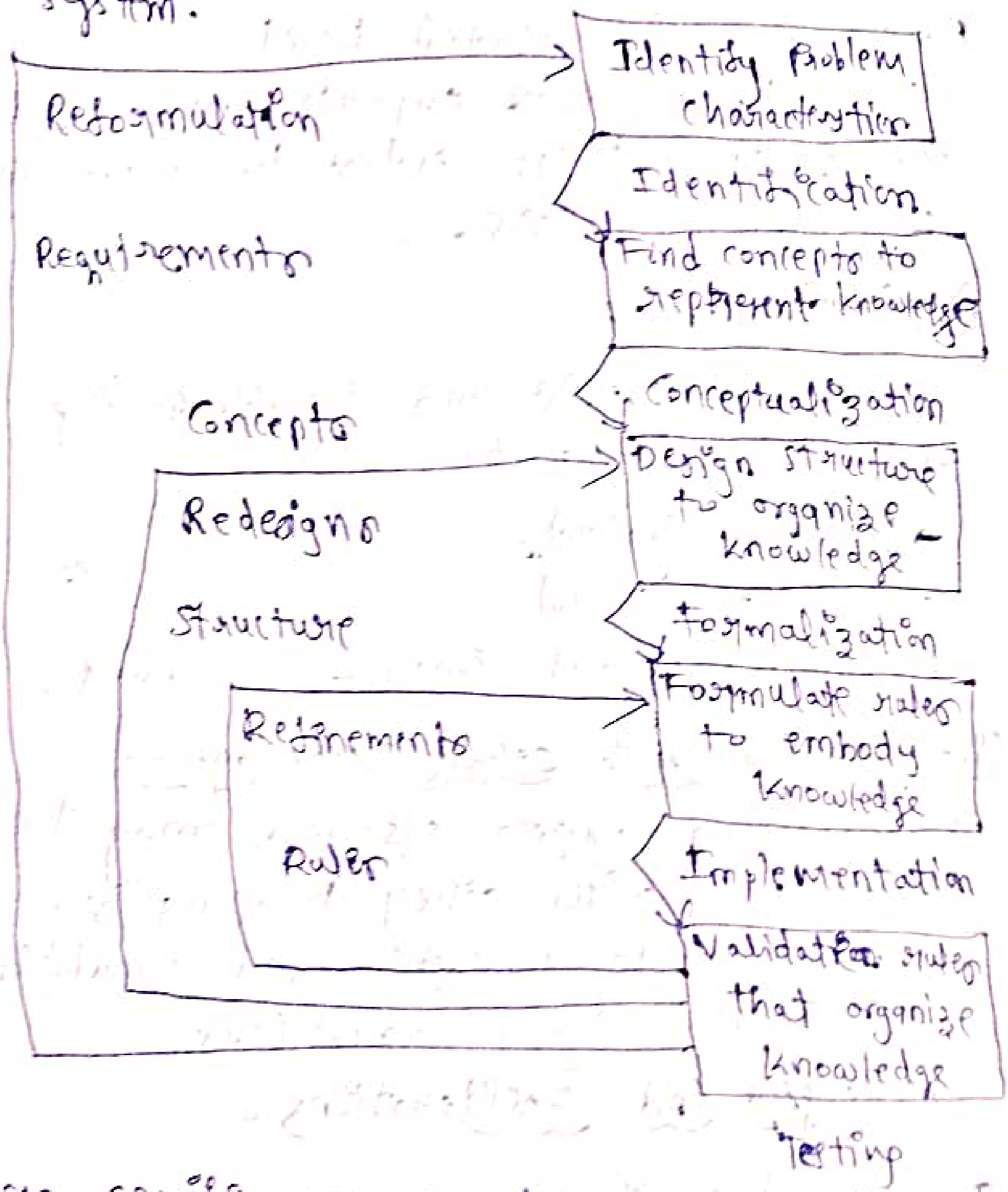
Most AI type problems at the level that many humans can deal with effectively are way beyond the current state of the art for practical SE.

In the Conceptualization stage, the task is to identify and explicate the key concepts, relations, and information flow patterns in the particular perspective on the problem identified in the previous stage.

The Formalization stage is self-explanatory, but we should note that formalization usually have a side-effect of introducing further constraints in our conceptualization of the proble

The Implementation stage amounts to formulating the rules that constitute the KB, and the control strategy to be used for reasoning over this KB.

Testing concerns evaluating the performance of the prototype and introducing modifications as indicated by the domain experts critique of the system.



The more specific reason why an expert system is never finished is two-fold:

(i) The KB is never complete.

(ii) There is often no absolutely correct answer for the system to produce.

Engineering expert systems play a crucial role in automating and assisting engineering tasks by leveraging domain-specific knowledge and logical reasoning.

These systems are designed to emulate the decision-making process of expert Engineers in various fields.

Rule-Based Reasoning:

Rules are formulated based on the knowledge provided by human experts, and the inference engine uses these rules to draw conclusions or make decisions.

Applications:

→ Fault diagnosis and troubleshooting
→ Design support.
→ Process Control
→ Quality control
→ Environmental impact assessment.

Integration with other Technologies:

Engineering expert systems may be integrated with other technologies, such as sensors, data analytics, and simulation tools, to enhance their capabilities.

Challenges and Considerations:

→ Knowledge maintenance
→ Handling uncertainty,
→ User interaction

The Lessons of Expert Systems for Engg AI s/w:-

An important cache of examples of AI s/w is to be found at the DEC- Digital Equipment Corp.

The R1 system checks customer orders of computer systems. It is designed to minimize the shipping out of wrong (or incompatible components.

Expert system technology was explicitly employed to addr an outstanding problem that a conventional s/w engg approach failed to solve.

The R1 system were easily quantifiable and the systems has now been successfully used for many years and a lot of empirical data has been collected and published.

Many expert systems have the ability to 'explain' their line of reasoning that led to a particular outcome.

Iterative system dev requires that someone can understand the specific system behavior in terms of general system structure.

The How constraint is necessary for system dev. It is also necessary for both validation and use of an adequate system.

The lessons learned from the dev and use of expert systems in engg can provide valuable insights for the creation of AI s/w in general.