Part IV — Defence Time: Mitigation Strategies

Based on the observations from testing local LLMs with Ollama (prompt injection, data poisoning, model inversion, and extraction patterns), several defensive strategies can be proposed to reduce risks and improve the security of local LLM deployments.

---

1. Input-Sanitisation Workflows
- **Purpose:** Prevent malicious or misleading prompts from influencing model behaviour.
- **Implementation Ideas:**
  - Strip or escape special instructions that attempt to override system prompts.
  - Detect suspicious patterns such as "ignore all instructions" or conflicting commands.
  - Enforce length, format, and content constraints on user inputs.

---

2. Output Verification Layers
- **Purpose:** Ensure the model's responses are safe, accurate, and within expected bounds.
- **Implementation Ideas:**
  - Use automated checks for sensitive information leakage.
  - Compare outputs against reference knowledge or trusted sources.
  - Flag or block responses that match known malicious patterns.

---

3. Rate-Limiting and Access Control
- **Purpose:** Prevent model extraction or abuse through repeated queries.
- **Implementation Ideas:**
  - Limit the number of requests per user or session.
  - Implement authentication and role-based access control.
  - Track unusual query patterns that may indicate extraction attempts.

---

4. Monitoring and Incident-Response Procedures
- **Purpose:** Detect and respond to attacks or anomalous behaviour.
- **Implementation Ideas:**
  - Log all user queries and model outputs for auditing.
  - Monitor for abnormal patterns (e.g., repeated instruction overrides).
  - Define a response plan for suspected model misuse or data leakage.

---

5. Governance, Compliance, and Documentation
- **Purpose:** Maintain accountability and regulatory compliance.
- **Implementation Ideas:**

  - Document model usage policies and security practices.
  - Define roles for model oversight and approvals for sensitive tasks.
  - Regularly review and update security guidelines.

---

6. Supply-Chain Verification of Model Sources
- **Purpose:** Ensure models and updates are trusted and untampered.
- **Implementation Ideas:**
  - Verify model checksums, signatures, or source repositories.
  - Avoid using unverified or unofficial models in production.
  - Maintain a version-controlled record of downloaded models.

---

7. Secure Fine-Tuning and Data-Handling Policies
- **Purpose:** Prevent internal data leaks and reduce poisoning risks.
- **Implementation Ideas:**
  - Use only verified and sanitized datasets for fine-tuning.
  - Encrypt and isolate sensitive data used in training or testing.
  - Review training pipelines to prevent inadvertent incorporation of malicious dat

🔹 Summary

By combining **input sanitisation**, **output verification**, **access controls**, and **supply-chain safeguards**, local LLM deployments can mitigate common security threats like prompt injection, data poisoning, model inversion, and extraction. Regular monitoring, documentation, and secure fine-tuning policies further strengthen the defense posture while maintaining privacy and reliability.