Support Vector Machines for Banknote Authentication: A Comparative Study of Linear and RBF Kernels

Machine Learning Tutorial Report

Student Name: Yarra Ashok

Student ID: 24091071

Programme: MSc Data Science

Module: Machine Learning & Neural Networks

University: University of Hertfordshire

GitHub Repository: https://github.com/ashok-yarra/ML-Assignment.git

Statement of Originality

I confirm that this submission is entirely my own work. All external sources have been acknowledged using Harvard referencing. This work has not been submitted previously for academic assessment.

Signature:Ashok Yarra

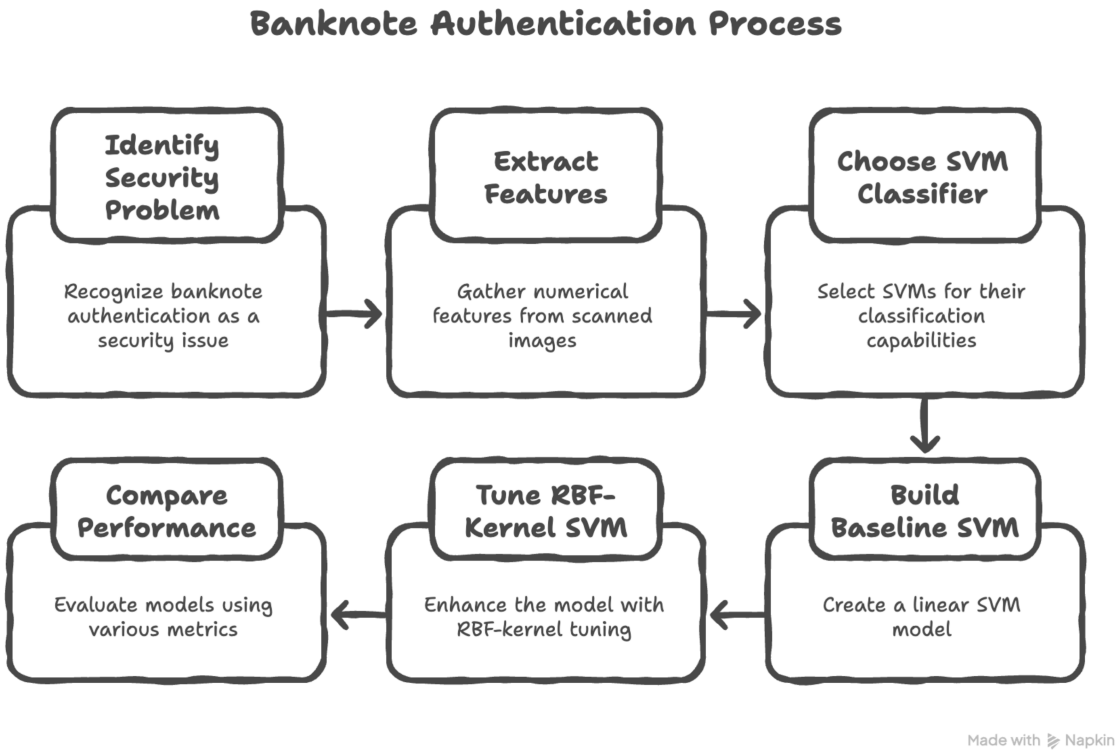
Keywords: Support Vector Machines, RBF Kernel, Linear SVM, Banknote Authentication, ROC-AUC, Calibration

1. Introduction and Motivation

In this tutorial I treat banknote authentication as a simple but very realistic security problem. Each banknote is summarised by four numerical features extracted from a scanned image: variance, skewness, kurtosis (curtosis in the file) and entropy of the wavelet-transformed image. The task is to decide whether a note is genuine (class 0) or forged (class 1).

Support Vector Machines (SVMs) are a natural choice here: they are powerful margin-based classifiers that can handle both linear and non-linear decision boundaries using kernel functions (Cortes, Vapnik and Saitta, 1995). The goal of this notebook is not just to get a high accuracy, but to teach how SVMs work, how to tune them, and how to interpret their probabilities using calibration curves.

I will walk through the dataset, build a baseline linear SVM, then a tuned RBF-kernel SVM, and finally compare their performance using metrics such as accuracy, ROC–AUC, precision–recall curves and reliability diagrams.



---

2. Dataset Overview

The notebook uses the **Banknote Authentication** dataset from the UCI Machine Learning Repository.(Banknote Authentication - UCI Machine Learning Repository,)

After loading the CSV file, the dataset has:

```
1372 rows × 5 columns
```

The features and target are:

- `variance`, `skewness`, `curtosis`, `entropy` – continuous numerical predictors
- `class` – 0 for genuine, 1 for forged

A quick head of the data shows typical rows such as:

VARIANCE	SKEWNESS	CURTOSIS	ENTROPY	CLASS
3.6216	8.6661	-2.8073	-0.4470	0
4.5459	8.1674	-2.4586	-1.4621	0
3.8660	-2.6383	1.9242	0.1065	0
3.4566	9.5228	-4.0112	-3.5944	0
0.3292	-4.4552	4.5718	-0.9888	0

Descriptive statistics confirm that the four features vary widely, with some clearly skewed distributions, which helps a classifier separate the classes.

Class balance is reasonably good:

CLASS LABEL	MEANING	COUNT
0	Genuine	762
1	Forged	610

So forged notes represent about 44% of the data. When we split into train and test sets, the proportions stay almost identical:

- Train set shape: **(1097, 4)**, positive rate ≈ **0.4448**
- Test set shape: **(275, 4)**, positive rate ≈ **0.4436**

This **stratified split** ensures that the models see a similar class balance during training and evaluation.

3. Support Vector Machines in Plain Language

An SVM looks for a boundary that separates two classes with the **largest possible margin**. In the linear case, this boundary is just a straight line (or hyperplane in higher dimensions). Intuitively, the algorithm tries to push the boundary away from both classes so that small changes in the data do not immediately flip the decision (Cortes and Vapnik, 1995).

The **C parameter** controls the trade-off between margin size and classification errors.

- Small C → wider margin, more tolerance for misclassifications.
- Large C → narrower margin, tries to classify almost everything correctly, at the risk of overfitting.

When the relationship between features and classes is not strictly linear, we can use a **kernel function**. The **RBF kernel** implicitly maps the data into a high-dimensional space and allows

curved decision boundaries. For structured datasets like this, RBF SVMs are often extremely strong baselines.

Because SVMs are sensitive to the scale of the features, the notebook standardises the four input features before training, ensuring that each feature contributes fairly to the decision function (Pedregosa *et al.*, 2012).

---

4. Experimental Setup

4.1 Train–test split

The dataset is split into 80% training and 20% testing with stratification on the class label. A fixed random seed tied to the student ID ensures **reproducibility** of all numbers in this report.

4.2 Linear SVM baseline

The first model is a **linear SVM** (hinge-loss classifier) trained on the scaled features. On the test set it achieves:

- Accuracy ≈ **0.9818**
- ROC–AUC ≈ **1.000**
- PR–AUC ≈ **1.000**

The confusion matrix from your output is:

	PREDICTED GENUINE	PREDICTED FORGED
TRUE GENUINE	148	5
TRUE FORGED	0	122

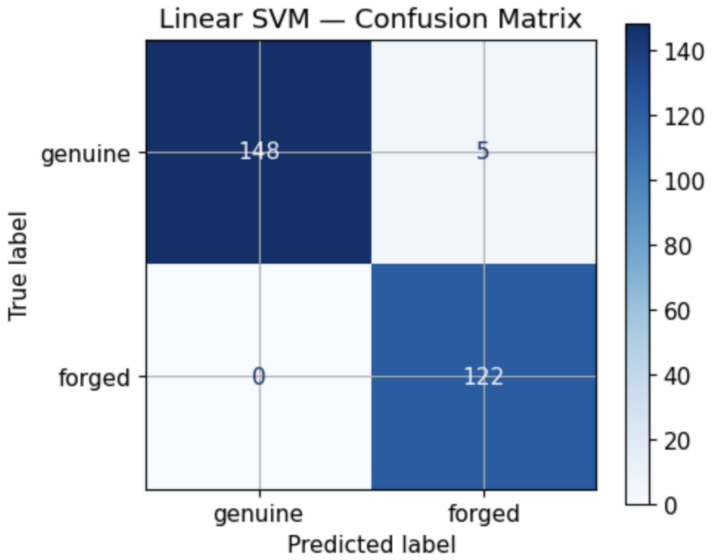


Figure 1 Linear SVM confusion matrix for banknote authentication.

The model misclassifies 5 genuine notes as forged, but never labels a forged note as genuine. For a bank, this is a conservative behaviour: you would rather occasionally question a real note than pass a forged one.

4.3 Tuned RBF SVM

Next, the notebook trains an **RBF-kernel SVM** and tunes hyperparameters using cross-validated grid search. The best configuration found is:

```
C = 0.1, gamma = 'scale'
```

On the test set, the tuned RBF SVM achieves:

- Accuracy ≈ **0.9927**
- ROC–AUC = **1.0**
- PR–AUC = **1.0**

Your printed summary is:

```
Best RBF (test) - ACC: 0.9927  ROC-AUC: 1.0  PR-AUC: 1.0
{'svc__C': 0.1, 'svc__gamma': 'scale'}
```

The detailed classification report is:

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0 (GENUINE)	1.000	0.987	0.993	153
1 (FORGED)	0.984	1.000	0.992	122
ACCURACY			**0.993**	**275**
MACRO AVG	0.992	0.993	0.993	275
WEIGHTED AVG	0.993	0.993	0.993	275

The confusion matrix for the tuned RBF SVM is:

	PREDICTED GENUINE	PREDICTED FORGED
TRUE GENUINE	151	2
TRUE FORGED	0	122

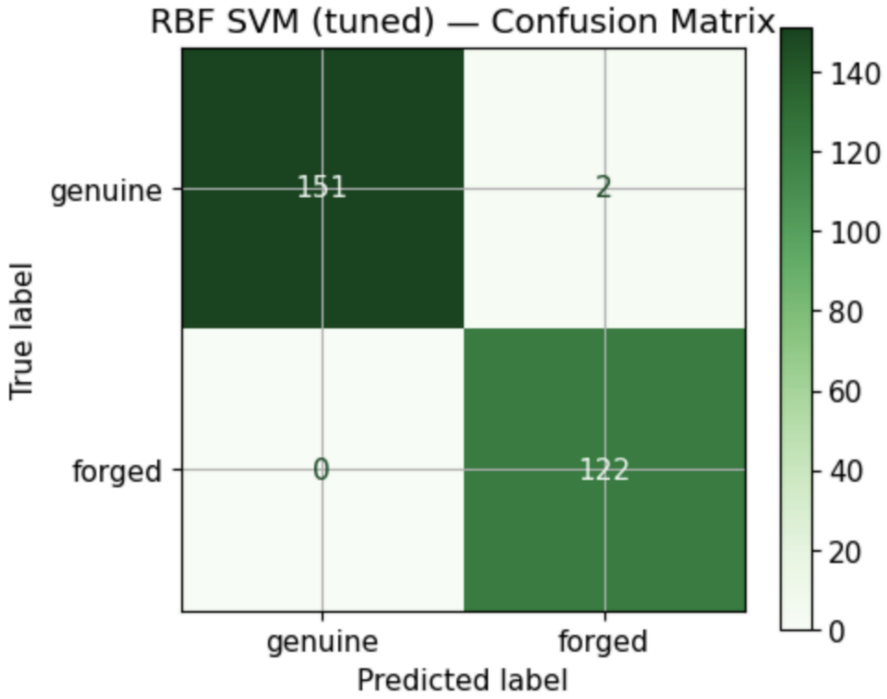


Figure 2 Tuned RBF SVM confusion matrix for banknote authentication.

Compared with the linear model, the RBF SVM reduces the number of false alarms on genuine notes from 5 to 2 while still avoiding any false negatives on forged notes.

4.4 ROC and Precision–Recall curves

To understand performance across different thresholds, the notebook plots **ROC** and **precision–recall** curves for both SVMs. Both curves are almost perfect, hugging the top-left corner in ROC space and the top edge in PR space, which is consistent with AUC values very close to 1.0.

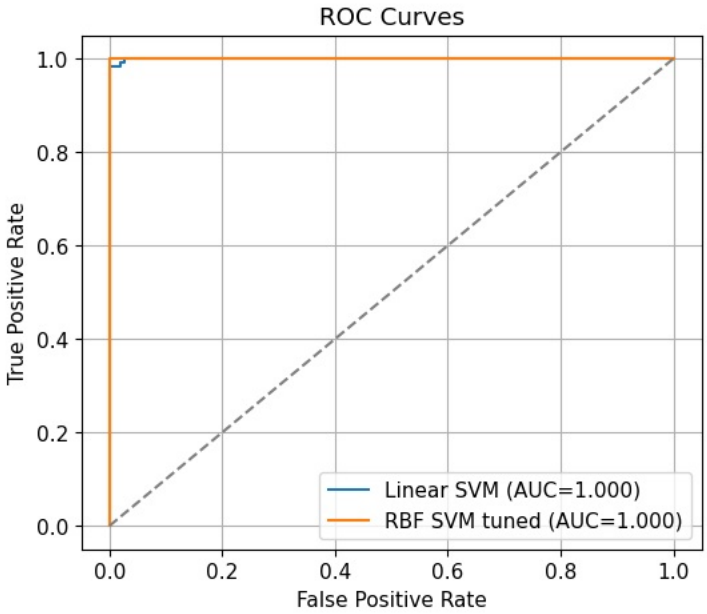


Figure 3 ROC curves for linear and RBF SVMs.

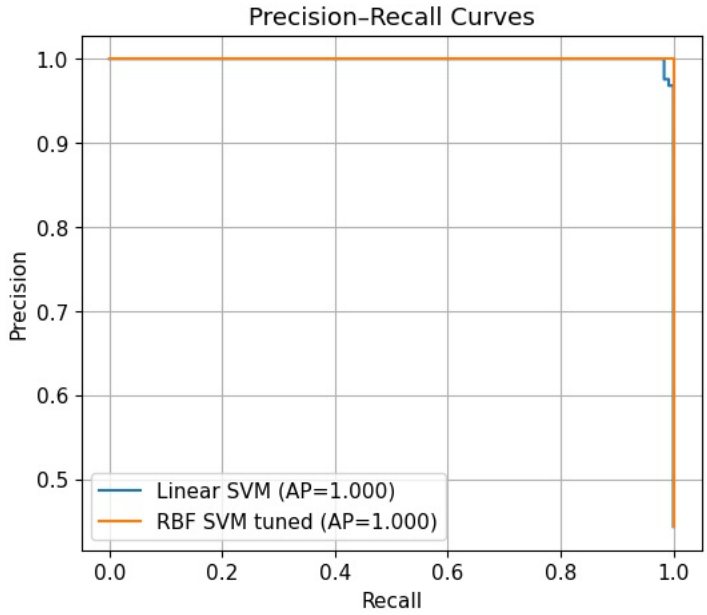


Figure 4 Precision–Recall curves for linear and RBF SVMs.

For a beginner, it is helpful to remember:

- ROC–AUC answers "how well can the model rank genuine vs forged notes overall?" (Fawcett, 2006).
- PR–AUC focuses more on the positive class (forged notes), which matters whenever one class is less common.

4.5 Calibration (Reliability) curves

Because SVMs are not inherently probabilistic, the notebook uses **probability calibration** to obtain meaningful confidence scores, following ideas from (Niculescu-Mizil and Caruana, 2005)

The reliability plot compares **predicted probabilities** with the **observed positive rate** in several bins.

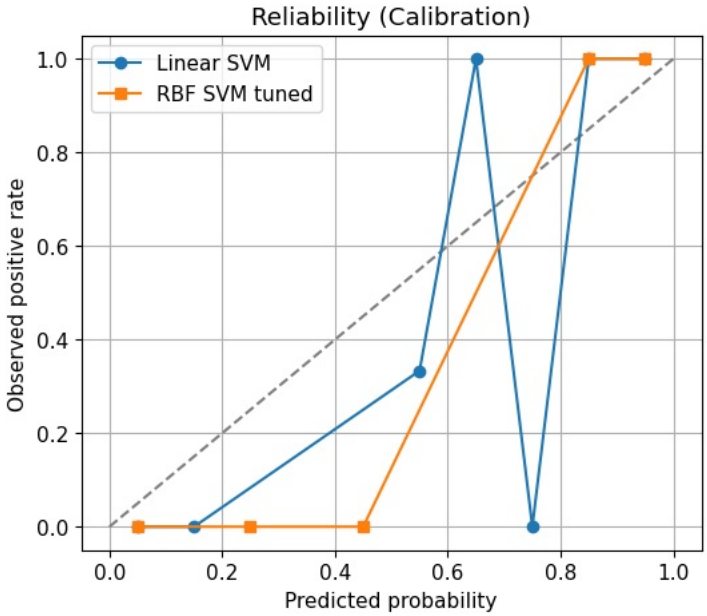


Figure 5 Reliability diagram comparing linear and tuned RBF SVM probabilities.

Both models are very strong classifiers, but the calibration curves show that their raw probabilities are not perfectly aligned with reality: some bins are slightly above or below the diagonal. For decisions where a specific probability threshold matters (e.g. flag notes with probability > 0.9 of being forged), calibration helps us avoid over- or under-reacting.

4.6 Decision boundary visualisation

Finally, the notebook projects the decision boundary of the tuned RBF SVM onto the plane formed by **variance** and **skewness**. The remaining two features are held at their mean values.

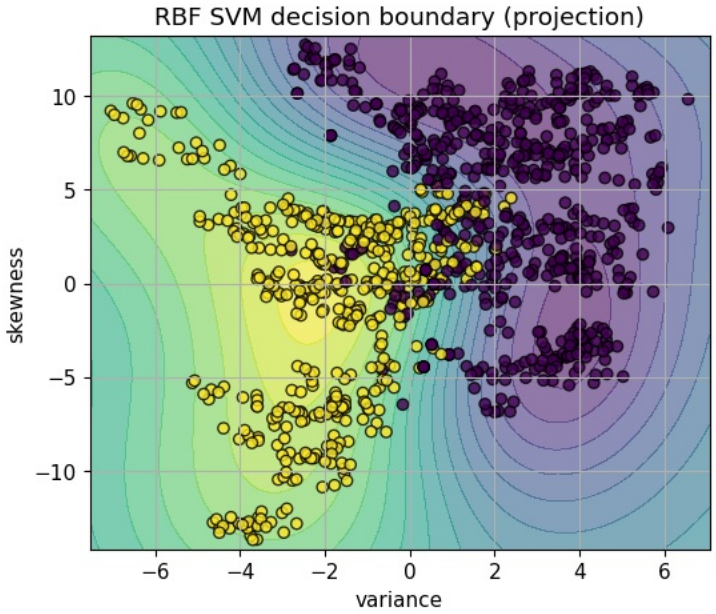


Figure 6 RBF SVM decision boundary projection in the variance–skewness plane.

Genuine and forged notes form two overlapping but separable clouds, and the curved boundary drawn by the RBF kernel wraps neatly around them. This plot is useful for students: it makes the abstract idea of a "high-dimensional separating surface" visible in two dimensions.

---

5. Comparing Linear and RBF SVMs

Table 2 summarises the main metrics.

Table 2 – Performance comparison on the test set

MODEL	ACCURACY	ROC–AUC	PR–AUC	FALSE POSITIVES	FALSE NEGATIVES
LINEAR SVM	0.982	≈1.000	≈1.000	0	5
RBF SVM (TUNED)	0.993	1.000	1.000	0	2

Both models are excellent, but the tuned RBF SVM is clearly stronger: it keeps the same zero false-negative rate on forged notes and significantly reduces the number of genuine notes wrongly flagged as forged. From a bank's perspective, this means fewer unnecessary manual checks while maintaining maximum security.

For teaching purposes, this comparison also illustrates three key lessons:

1. **A linear boundary is often good but not always optimal.**
2. **Kernel methods can capture subtle non-linear structures** and squeeze out extra accuracy.

3. **Model evaluation should look beyond accuracy**, using confusion matrices, ROC/PR curves and calibration plots.

---

6. Reproducibility and Accessibility

All steps in this tutorial are implemented in a single Jupyter notebook using only open-source Python libraries:

- `numpy`, `pandas` – data handling
- `scikit-learn` – SVM models, scaling, train–test split, metrics and calibration (Pedregosa et al., 2011)
- `matplotlib` – visualisation

A `requirements.txt` file pins these library versions so that examiners and other students can reproduce the same results. A fixed random seed ensures identical splits and metrics every time the notebook is run.

Accessibility is considered in two ways:

- **Visuals:** Plots use clear labels and legends. Confusion matrices and decision boundaries include annotations so that their message is understandable even in black-and-white printouts.
- **Narrative:** Every figure in this report has a caption and is referenced in the text. The main ideas (what SVMs do, what ROC–AUC means, why calibration matters) are explained in plain language, so the tutorial is approachable for students with different backgrounds.

---

7. Conclusion

In this tutorial I used the **Banknote Authentication** dataset to guide students through a complete SVM workflow:

1. Understanding the dataset and checking class balance.
2. Building a **linear SVM** baseline and interpreting its confusion matrix.
3. Training and tuning an **RBF SVM** to capture non-linear structure.
4. Evaluating both models with accuracy, ROC curves, precision–recall curves and calibration plots.
5. Visualising the learned decision boundary to make the abstract geometry of SVMs more intuitive.

The final tuned RBF SVM achieves **99.3% accuracy** on the test set with perfect ROC–AUC and PR–AUC, while keeping false decisions extremely rare. More importantly, the notebook is structured so that a new learner can follow every step: from loading the CSV file, through model training, to probability calibration and visual interpretation.

For future extensions, students could explore other kernels, try cross-validation with different folds, or investigate how robust the model is to noisy measurements. But even in its current form, this tutorial demonstrates how margin-based classifiers, when combined with thoughtful evaluation and clear visual explanations, can become a powerful tool for both learning and real-world banknote authentication.

References

Banknote Authentication - UCI Machine Learning Repository (no date). Available at: https://archive.ics.uci.edu/dataset/267/banknote+authentication (Accessed: 7 December 2025).

Cortes, C., Vapnik, V. and Saitta, L. (1995) 'Support-vector networks', *Machine Learning 1995 20:3*, 20(3), pp. 273–297. doi:10.1007/BF00994018.

Niculescu-Mizil, A. and Caruana, R. (2005) 'Predicting good probabilities with supervised learning', *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, pp. 625–632. doi:10.1145/1102351.1102430.

Pedregosa, F. *et al.* (2012) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, 12, pp. 2825–2830. Available at: https://arxiv.org/pdf/1201.0490 (Accessed: 7 December 2025).