

# Expert Banking System for Fraud Detection and Loan Approval Analysis

Ali Hashish<sup>1</sup>, Mostafa Osama<sup>2</sup>, and Omar M. Amin<sup>3</sup>

Computer Engineering Department, Faculty of Engineering, Cairo University, Egypt

This paper focuses on two critical applications in the banking sector: loan approval analysis and fraud detection. As financial institutions increasingly adopt data-driven solutions, it is important to evaluate whether traditional knowledge-based expert systems can still provide value alongside machine learning and deep learning approaches. We present a comparative study using publicly available datasets by constructing a banking expert system based on decision trees generated using the ID3 algorithm and benchmarking it against supervised models such as Support Vector Machines (SVM) and Logistic Regression. Additionally, we compare our results with findings from existing literature where ensemble models like AdaBoost and XGBoost were applied to similar problems. Our results show that rule-based expert systems can offer competitive performance and may serve as practical alternatives to machine learning models.

**KEYWORDS:** Expert Systems, Knowledge Base, Machine Learning, Credit Risk Management, Credit Analysis, Banking

## 1 INTRODUCTION

In a society centered on finance, banking systems inevitably become crucial aspects of every person's life. The increase in demand for services provided by banks necessitates an efficient yet highly accurate approach to automate said services whenever possible. Our paper aims to tackle this issue by implementing a banking expert system that detects fraudulent transactions and facilitates loan approval analysis.

While most researchers head towards utilizing machine learning and deep learning techniques due to the abundance of data for every problem, many fail to consider the limitations of these approaches, and whether better alternatives exist. These kinds of techniques are usually computationally intensive, and require highly capable GPUs to operate smoothly. Companies circumvent this cost by opting to deploy their products on

---

<sup>1</sup>[ali.hashish001@gmail.com](mailto:ali.hashish001@gmail.com)

<sup>2</sup>[mostafa.mahmoud00@eng-st.cu.edu.eg](mailto:mostafa.mahmoud00@eng-st.cu.edu.eg)

<sup>3</sup>[oamin.omar@gmail.com](mailto:oamin.omar@gmail.com)

cloud servers, but this also comes at a great environmental cost, as explained by Steven Monserrate [13].

On the other hand, expert systems are a lot cheaper to maintain, as they require much less resources to run. Through our paper, we aim to prove that traditional expert systems are still quite valuable, and can even outperform modern machine learning approaches in some cases. We build our rules using the ID3 algorithm [14], and compare it to an SVM [7] model and a logistic regression [11] model. We also compare our findings with some ensemble models like AdaBoost [theoretic'1997] and XGBoost [chen'xgboost]. We use publicly available datasets for the evaluation of models, namely the loan analysis dataset provided by Archit Sharma [15] and the fraud detection dataset provided by Samay Ashar et al [3]. Our code can be easily viewed through our GitHub repository<sup>4</sup>.

The expert systems show promising results, surpassing all other approaches for the loan approval analysis, with the exception of the XGBoost, that scores slightly higher than the rule-based approach's 96.83% accuracy. It also achieves the highest precision of 99.3%, a recall of 92.26% and an F1-score of 95.67%. As for the fraud detection problem, the expert system scores higher than all other approaches, achieving a perfect score.

To concisely summarize, our contributions lie primarily in building an expert system and showing that it is as capable as other advanced machine learning techniques, and more cost-effective due to its comparative simplicity.

The paper outline is as follows: Firstly, Section 2 discusses the literature surrounding expert systems and machine learning techniques with respect to banking systems. Secondly, Section 3 thoroughly explains the pipeline of our project, showing how we obtain the dataset and preprocess it. It also details the process of building the expert system. Followed by that, Section 4 contains the results and insights regarding them. Finally, Section 5 concisely concludes the paper, highlighting our future work.

## 2 RELATED WORK

Knowledge-based Expert Systems have long been a key area of research, especially for their integration into real-world applications. Many of these applications have focused on the banking sector and Compliance Auditing (CA).

Some earlier studies, such as K Bryant's [4], focused on agricultural loan evaluation by combining both qualitative and quantitative assessment methods. Mostafa Mahmoud et al [10] focused on the credit-risk evaluation problem, which is a critical issue in the

---

<sup>4</sup>Repository link: [Expert Banking System](#)

banking and financial sector. Earlier approaches to this problem typically relied on more traditional statistical methods.

With the continuous advancement of expert system techniques and the improved results they have demonstrated over time, banks have gradually developed greater trust in expert systems and digital technologies. This shift aims to enhance productivity, reduce human error, and support more objective, fact-based decision-making. As a result, more recent studies have begun leveraging large data sources to build more powerful knowledge bases and have incorporated modern machine learning techniques into their systems. Researches like Ali et al [2] tackled the problem of bank customer credit scoring evaluation using Fuzzy Expert System which uses membership functions, defined by domain experts, to map numeric inputs to categorical output using fuzzy logic. Mustafa Menekay et al [12] also applied a Fuzzy Expert System for bank credit authorization, which was used in financial credit evaluation and approval processes.

Moreover, researchers are leveraging modern Machine Learning techniques for financial and bank-related problems. F. M. Ahosanul et al [6] worked on loan approval application where the researchers compared the following algorithms: **Decision Tree Categorization**, **Adaptive Boosting (AdaBoosting)** [5], **Random Forest Classifier**, **Support Vector Machine (SVM)** [7], and **Gaussian Naive Bayes (GaussianNB)**.

In our research, we decided to tackle two major problems banks and financial institutions face on a daily basis: **Loan Approval Analysis** and **Fraud Detection**. To illustrate, while modern applications tend to integrate machine learning models to solve these kinds of problems, we decided to develop **Knowledge-Based Expert Systems** and compare the results with supervised machine learning algorithms: **SVM** [7] and **Logistic Regression** [11], as well as **XGBoost** [chen'xgboost] and **AdaBoost**[5]. We used the **Iterative Dichotomizer 3 (ID3)** Decision Tree algorithm [14] to generate the necessary rules. For the inference engine, we used **Experta** [1], a Python-based library for building expert systems, inspired by the **CLIPS** Rule-Based Programming Language.

### 3 PROPOSED APPROACH

For our banking system, we use a number of different techniques to help us detect transaction fraud, and analyze loan eligibility. We obtain the data from publicly available sources such as Kaggle. For the transaction fraud, we use the dataset prepared by Samay Ashar et al [3], as for the loan eligibility, we use the one provided by Archit Sharma [15]. We will be thoroughly explaining our approach in this upcoming section.

### 3.1 Loan Approval Analysis

#### 3.1.1 Dataset Preparation

The loan approval analysis dataset is comprised of 4,269 records, of which 2,656 records are labeled as 0 for approved while 1,613 are 1 for rejected, as explained in figure 3.1.1.

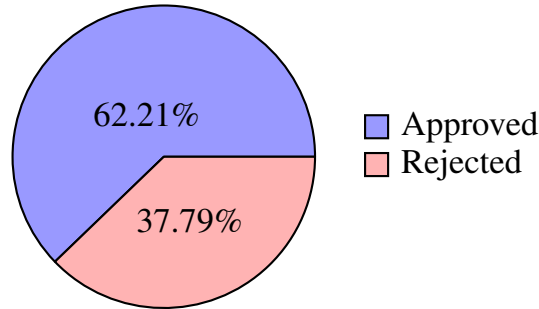


Figure 1: Distribution of Loan Target Class in The Dataset

The dataset is made up of 13 columns containing all the necessary data for accurately predicting the label. The data revolves around the loan-applier, their income, loan amount, assets and credit score.

Naturally the first step is to preprocess the input data. We notice our data is devoid of null values, so we move on to the next step: label encoding. We encode our only non-numeric field, the education, into 0 and 1 for graduate and not graduate respectively.

Having ensured all data is numerical, we now need to normalize the fields, due to the great variety in their ranges. For instance, most asset-related fields are in the millions, while loan term and credit score are in the tens and hundreds. To achieve a better balance, we use the min-max scaler to normalize the data, ensuring all fields range from 0 to 1. We then proceed to simply splitting the data into training set and testing set with a ratio of 80:20 respectively.

#### 3.1.2 Rule Based Approach - Decision Tree

In order to analyze the loan status, we utilize a few methods, ranging from building an expert system, to using popular machine learning techniques such as SVM [7] and logistic regression [11]. For our expert system, we depend on the ID3 algorithm [14] to build the rules.

We experimented with the tree parameters, settling on the following:

1. **Criterion:** The criterion used by the tree to split that data on. We use entropy as our criterion.

2. **Maximum depth:** The maximum allowed depth of the tree. We settled on 4, to prevent the tree from being too deep and creating too many rules.
3. **Minimum sample split:** The minimum number of samples needed in a node to consider splitting it. We set a minimum sample split of 10, in order to avoid overfitting and getting nodes with so few examples.
4. **Minimum samples leaf:** The minimum number of samples required in a leaf node. We pass it as 5 after trying out a few different values.
5. **Minimum impurity decrease:** The minimum decrease in entropy required for a split to take place. We settle on 0.001 to avoid meaningless splits.

Through these parameters, we produce the decision tree displayed in figure 2. The tree has a depth of 4, and uses a number of vital features to predict the loan status. The format of the rules is changed, using a simple python script, to match the format expected by Experta [1]. **Experta** is a python library designed to facilitate building expert systems. With this step, our knowledge base is complete, we can now make use of **Experta**'s inference engine to predict the loan status.

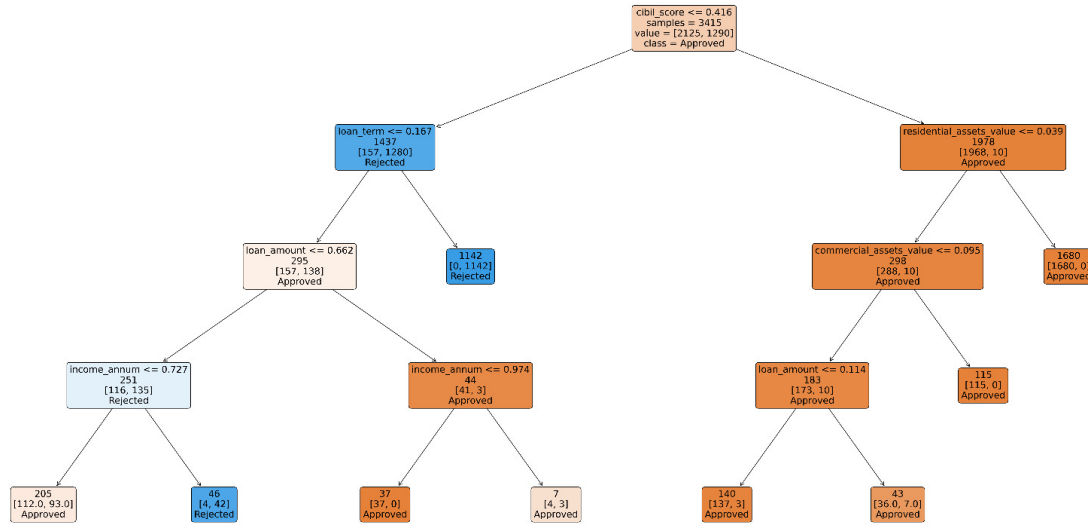


Figure 2: Rules generated using the decision tree on loan approval analysis dataset

### 3.1.3 Machine Learning Techniques

We add an option to use machine learning techniques instead of an expert system. The preprocessing steps are shared with the previously explained steps in section 3.1.1, as

well as the splitting ratio and random seed. This setup ensures reproducibility, making the results produced more reliable when comparing across models.

We train an SVM [7] model and a logistic regression [11] model.

- For the **SVM** model, to draw the decision boundaries, we use a polynomial kernel of the fourth degree. We pass the regularization parameter as 4, in an attempt to obtain better generalization through relatively wide margins, and to avoid overfitting.
- For the **Logistic Regression** model, we bound the maximum iterations to 1,000 so that the model exits even if it does not converge in those iterations. We employ a liblinear algorithm, as it is generally a good choice for binary classification. We set the inverse of regularization strength to 1, to ensure moderate regularization. Finally, for the penalty, we use the L2 ridge regularization.

## 3.2 Fraud Detection

### 3.2.1 Dataset Preparation

The fraud detection dataset consists of 50,000 records labeled as 0 for not fraud, and 1 for fraud. The number of non-fraud records is slightly more than double that of the fraud, as shown in figure 3.2.1. That is to be expected, as naturally the number of fraudulent transactions is much lower than that of real transactions.

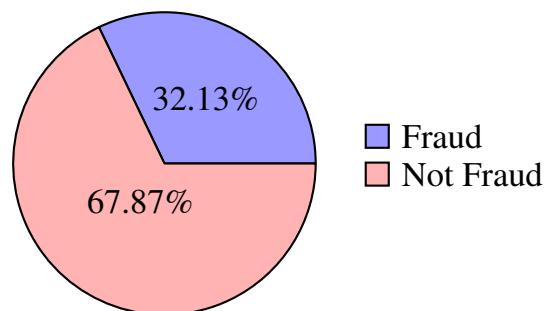


Figure 3: Distribution of Fraud Target Class in The Dataset

The dataset contains crucial data for determining the authenticity of a transaction, such as transaction amount and type, account balance, statistics for card usage, risk score, and transaction timestamp, as well as other equally salient features.

We start our preprocessing stage by examining if there exists any null values, in this case there were none. Accordingly, we proceed to label encode the non-numeric fields, such as the transaction type and card type, ensuring all fields become numerical.

A quick look at the table easily reveals the great discrepancies between the ranges of the fields; account balance is orders of magnitude higher than daily transaction count for instance. We resolve this by using a min-max scaler to normalize the data, guaranteeing no one field can have too much effect on the outcome of the system. Finally, we split the data into training set and testing set with a ratio of 80:20

### 3.2.2 Rule Based Approach - Decision Tree

We use a number of approaches to classify fraudulent activity, such as building an expert system, or using common machine learning techniques. Similar to before, we use the ID3 algorithm in building the rules.

We tried different tree parameters, and these were the best ones:

1. **Criterion:** We use entropy as our criterion.
2. **Maximum depth:** We set it at 4, to prevent the tree from being too deep and creating too many rules.
3. **Minimum sample split:** We choose a minimum sample split of 100, in order to avoid overfitting and getting nodes with so few examples.
4. **Minimum samples leaf:** We set it as 50 to avoid having unnecessarily small leaves.
5. **Minimum impurity decrease:** We settle on 0.01 to focus on meaningful splits.

Using the aforementioned parameters, we obtain the simple decision tree shown in figure 4. It contains three rules covering the whole dataset. We use a simple python script to convert the rules to the format the expert system library **Experta** [1] requires. After building the knowledge base, we utilize **Experta's** inference engine to predict the labels.

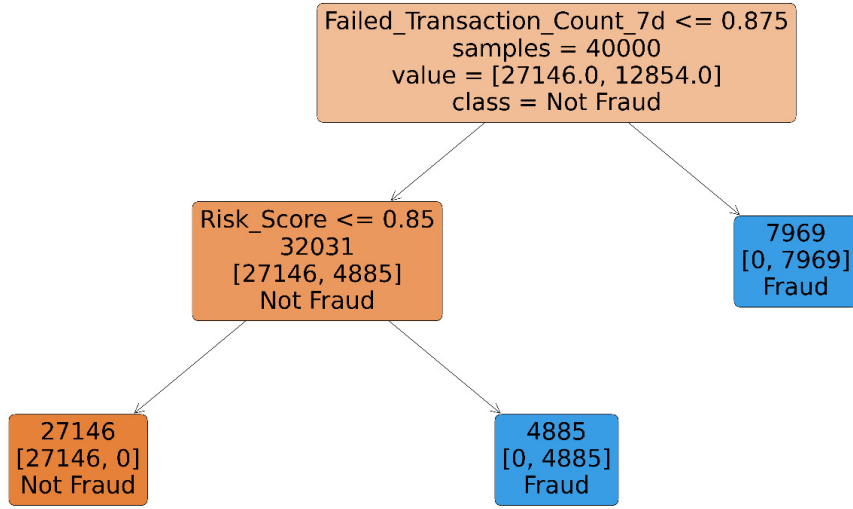


Figure 4: Rules generated using the decision tree on fraudulent dataset

### 3.2.3 Machine Learning Techniques

In addition to using expert systems, we also implement a number of common machine learning approaches. The preprocessing steps are identical to the steps mentioned previously in section 3.2.1 followed by the same dataset splitting ratio, and same random seed to guarantee reproducibility.

We test an SVM [7] model and a logistic regression [11] model.

- For the **SVM** model, we use a polynomial kernel of the fourth degree to find the decision boundaries. We set the regularization parameter to 4, thus obtaining better generalization with relatively wide margins.
- For the **Logistic Regression** model, we set the maximum iterations to 1,000 to prevent the model from getting stuck in case it does not converge. We use a liblinear algorithm, as it is generally a good choice for binary classification. We fix the inverse of regularization strength to 1, to ensure moderate regularization. Finally, for the penalty, we use the L2 ridge regularization.

Thus, the outline of our pipeline is as shown in figure 5. We determine whether to make a loan approval analysis, or detect fraud, then we choose one of the available approaches, whether using expert systems or a machine learning technique as SVM or logistic regression.



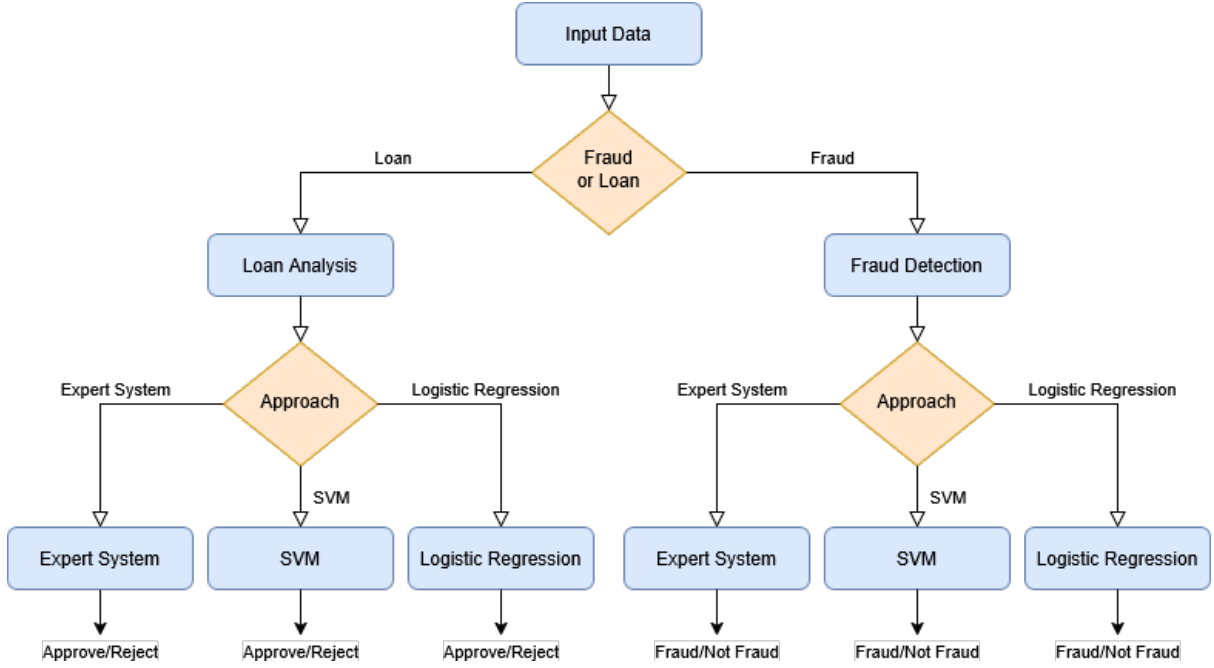


Figure 5: Banking system pipeline

## 4 RESULTS

Our main goal was to show that expert systems can still perform well in many domains without relying on machine learning or deep learning approaches. To support this hypothesis, we compare the prediction Accuracy, Precision, Recall, and F1-score of SVM and logistic regression models with the results from the rules generated by our knowledge base.

### 4.1 Evaluation Metrics

To compare our generated rules with the other machine learning approaches, we use four main evaluation metrics:

- **Accuracy:** The percentage of total predictions that were correct, indicating the overall correctness of the model.
- **Precision:** The percentage of correctly predicted positives out of all predicted positives, indicating how reliable the model is when predicting a positive case.
- **Recall:** The percentage of correctly predicted positives out of all actual positives, indicating the model's ability to identify positive cases.

- **F1-Score:** The harmonic mean of precision and recall, used to evaluate the balance between precision and recall, especially when both are equally important, as in our study.

#### 4.1.1 Loan Approval Analysis Results

For loan approval analysis, we used the ID3 algorithm to generate the rules shown in Table 1:

Table 1: Loan Approval Analysis Learned Rules

Rule	Conditions	Loan Status
0	$CS \leq 0.41583, LT \leq 0.16667, LA \leq 0.66199, IA \leq 0.7268$	Approved
1	$CS \leq 0.41583, LT \leq 0.16667, LA \leq 0.66199, IA > 0.7268$	Rejected
2	$CS \leq 0.41583, LT \leq 0.16667, LA > 0.66199, IA \leq 0.97423$	Approved
3	$CS \leq 0.41583, LT \leq 0.16667, LA > 0.66199, IA > 0.97423$	Approved
4	$CS \leq 0.41583, LT > 0.16667$	Rejected
5	$CS > 0.41583, RA \leq 0.03938, CA \leq 0.09536, LA \leq 0.11352$	Approved
6	$CS > 0.41583, RA \leq 0.03938, CA \leq 0.09536, LA > 0.11352$	Approved
7	$CS > 0.41583, RA \leq 0.03938, CA > 0.09536$	Approved
8	$CS > 0.41583, RA > 0.03938$	Approved

Abbreviations: *CS* = CIBIL Score, *LT* = Loan Term, *LA* = Loan Amount, *IA* = Income per Annum, *RA* = Residential Assets Value, *CA* = Commercial Assets Value

As shown in Table 2, the rule-based approach outperformed SVM and Logistic Regression in terms of accuracy, precision, and F1-score. We also compared it to the results of AdaBoost and XGBoost from the work of Honghao Yu[8]. The rule-based approach produced results comparable to AdaBoost, while XGBoost achieved the best performance overall. However, the difference in results between XGBoost and the rule-based approach was not substantial. moreover, rule-based approach is faster.

Table 2: Comparison of Evaluation Metrics of Loan Approval Analysis

Metric	Rule-Based	SVM	Logistic Regression	AdaBoost	XGBoost
Accuracy	0.9683	0.9449	0.9215	0.9672	<b>0.9824</b>
Precision	<b>0.9930</b>	0.9233	0.9238	0.9672	0.9824
Recall	0.9226	0.9319	0.8638	0.9672	<b>0.9824</b>
F1 Score	0.9567	0.9276	0.8928	0.9672	<b>0.9824</b>

### 4.1.2 Fraud Analysis Results

For fraud detection, we generated the rules based on patterns in failed transaction counts and risk scores as the data-set showed clear fraud patterns, as shown in Table 3:

Table 3: Fraud Analysis Learned Rules

Rule	Conditions	Fraud Status
0	$FT \leq 0.875, RS \leq 0.85004$	Not Fraud
1	$FT \leq 0.875, RS > 0.85004$	Fraud
2	$FT > 0.875$	Fraud

*Abbreviations: FT = Failed Transaction Count (7 days), RS = Risk Score*

As shown in Table 4, the rule-based approach achieved perfect scores in all evaluation metrics, outperforming both SVM and Logistic Regression and XGBoost model from Kong Kai Feng’s work [9] models in fraud detection performance.

Table 4: Comparison of Evaluation Metrics of Fraud Analysis

Metric	Rule-Based	SVM	Logistic Regression	XGBoost
Accuracy	<b>1.0000</b>	0.9854	0.7994	0.9994
Precision	<b>1.0000</b>	0.9794	0.7063	0.9984
Recall	<b>1.0000</b>	0.9751	0.6430	0.9997
F1 Score	<b>1.0000</b>	0.9772	0.6732	0.9991

### 4.1.3 Summary and Discussion

As shown in the loan approval analysis, the rule-based approach outperformed our machine learning models (SVM and Logistic Regression), achieving performance comparable to AdaBoost and XGBoost results reported in Honghao Yu’s Work [8]. In the fraud analysis, it achieved a perfect score; however, this may be related to a known limitation of the ID3 algorithm, which is based on decision trees and is prone to overfitting.

## 5 CONCLUSION

In this paper, we show the effectiveness of knowledge-based expert systems in addressing two critical banking challenges: loan approval and fraud detection. Despite the growing dominance of machine learning and deep learning, our study shows that expert systems, when carefully designed using robust rule generation techniques like ID3 can

offer competitive, and in some cases superior, performance across different evaluation metrics.

For loan approval analysis, the rule-based approach outperformed both the SVM and logistic regression models that we built, and achieved results comparable to more advanced techniques like AdaBoost and XGBoost implemented in Honghao Yu's work [8]. In the fraud detection task, the rule-based expert system achieved perfect scores across all evaluation metrics, surpassing all compared models. While this suggests high predictive power, it also highlights the potential risk of overfitting, a common limitation of decision tree-based rule generation.

Our findings suggest that expert systems, when provided with appropriate rule learning algorithms and well-prepared knowledge base, remain a viable and efficient alternative to machine learning models. This is especially valuable in domains like banking, where explainability and speed are essential.

As future work, we plan to extend our expert system to cover wider decision-making tasks within banking. This includes building rule-based solutions for customer behavior prediction, such as identifying which customers are more likely to participate in sales or marketing campaigns, or who are more willing to purchase or subscribe to various financial services. Such extensions would further validate the sufficiency of expert systems to meet different banking needs.

## REFERENCES

- [1] Roberto Abdelkader and David Francos. *GitHub - nilp0inter/experta: Expert Systems for Python*. en. Mar. 2025. URL: <https://github.com/nilp0inter/experta> (visited on 05/24/2025).
- [2] Soheila Sardar Donighi Ali Bazmara. "Bank customer credit scoring by using fuzzy expert system". en. In: *International Journal of Intelligent Systems and Applications* 6.11 (Oct. 2014), pp. 29–35. DOI: [10.5815/IJISA.2014.11.04](https://doi.org/10.5815/IJISA.2014.11.04). URL: <https://scispace.com/papers/bank-customer-credit-scoring-by-using-fuzzy-expert-system-4o5t40pp55> (visited on 05/24/2025).
- [3] Samay Deepak Ashar. *Fraud Detection Transactions Dataset*. en. 2025. URL: <https://www.kaggle.com/datasets/samayashar/fraud-detection-transactions-dataset> (visited on 05/24/2025).

- [4] K Bryant. “ALEES: an agricultural loan evaluation expert system”. In: *Expert Systems with Applications* 21.2 (Aug. 2001), pp. 75–85. ISSN: 0957-4174. DOI: [10.1016/S0957-4174\(01\)00028-8](https://doi.org/10.1016/S0957-4174(01)00028-8). URL: <https://www.sciencedirect.com/science/article/pii/S0957417401000288> (visited on 05/24/2025).
- [5] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of Computer and System Sciences* 55.1 (Aug. 1997), pp. 119–139. ISSN: 0022-0000. DOI: [10.1006/jcss.1997.1504](https://doi.org/10.1006/jcss.1997.1504). URL: <https://www.sciencedirect.com/science/article/pii/S002200009791504X> (visited on 05/24/2025).
- [6] F. M. Ahosanul Haque and Md Mahedi Hassan. *Bank loan prediction using machine learning techniques*. arXiv:2410.08886. Oct. 2024. DOI: [10.48550/arXiv.2410.08886](https://doi.org/10.48550/arXiv.2410.08886). URL: <http://arxiv.org/abs/2410.08886> (visited on 05/24/2025).
- [7] M.A. Hearst et al. “Support vector machines”. In: *IEEE Intelligent Systems and their Applications* 13.4 (July 1998), pp. 18–28. ISSN: 2374-9423. DOI: [10.1109/5254.708428](https://doi.org/10.1109/5254.708428). URL: <https://ieeexplore.ieee.org/abstract/document/708428> (visited on 05/24/2025).
- [8] hong haoyu hong. *Preprocessing+PrecisionRecall+Logistic/Voting/XGB*. en. 2025. URL: <https://kaggle.com/code/honghaoyu348/preprocessing-precisionrecall-logistic-voting-xgb> (visited on 05/25/2025).
- [9] kong kaifeng kong. *Fraud Detection by EDA analysis and XGBoost*. en. 2025. URL: <https://kaggle.com/code/kongkaifeng/fraud-detection-by-eda-analysis-and-xgboost> (visited on 05/25/2025).
- [10] Mostafa Mahmoud, Najla Algadi, and Ahmed Ali. “Expert system for banking credit decision”. In: *2008 International Conference on Computer Science and Information Technology*. Aug. 2008, pp. 813–819. DOI: [10.1109/ICCSIT.2008.31](https://doi.org/10.1109/ICCSIT.2008.31). URL: <https://ieeexplore.ieee.org/abstract/document/4624980/keywords#keywords> (visited on 05/24/2025).
- [11] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. en. Boston, MA: Springer US, 1989. ISBN: 9780412317606 9781489932426. DOI: [10.1007/978-1-4899-3242-6](https://doi.org/10.1007/978-1-4899-3242-6). URL: <http://link.springer.com/10.1007/978-1-4899-3242-6> (visited on 05/24/2025).

- [12] Mustafa Menekay and Mohammad Khalleel Sallam Ma'aitah. "Applying expert system for bank credit authorization using fuzzy tools". en. In: *Proceedings of the 2017 9th International Conference on Education Technology and Computers*. Barcelona Spain: ACM, Dec. 2017, pp. 258–261. ISBN: 9781450354356. DOI: [10.1145/3175536.3175585](https://doi.org/10.1145/3175536.3175585). URL: <https://dl.acm.org/doi/10.1145/3175536.3175585> (visited on 05/24/2025).
- [13] Steven Gonzalez Monserrate. "The Cloud Is Material: On the Environmental Impacts of Computation and Data Storage". en. In: *MIT Case Studies in Social and Ethical Responsibilities of Computing* Winter 2022 (Jan. 2022). DOI: [10.21428/2c646de5.031d4553](https://doi.org/10.21428/2c646de5.031d4553). URL: <https://mit-serc.pubpub.org/pub/the-cloud-is-material/release/2> (visited on 05/25/2025).
- [14] Edward Ogheneovo and Promise Nlerum. "Iterative Dichotomizer 3 (ID3) Decision Tree: A Machine Learning Algorithm for Data Classification and Predictive Analysis". In: *International Journal of Advanced Engineering Research and Science* 7 (Jan. 2020), pp. 514–521. DOI: [10.22161/ijaers.74.60](https://doi.org/10.22161/ijaers.74.60).
- [15] Archit Sharma. *Loan-Approval-Prediction-Dataset*. en. 2023. URL: <https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset> (visited on 05/24/2025).