UNIVERSITY COLLEGE ROOSEVELT
Lange Noordstraat 1, Middelburg, The Netherlands

Bachelor of Liberal Arts & Sciences

SENIOR PROJECT

Predicting Bank Failures Using Convolutional Neural Networks

An Exploratory Analysis

Author:
Michael Antony Rayan

Student Number:
1904388

Supervisor:
Dr. M. Jansen

Second examiner:
Dr. A. Karas

November 4, 2025

Abstract

This study investigates the application of Convolutional Neural Networks (CNNs) in predicting bank failures using a dataset of Russian banks spanning from 2004 to 2020. The problem is framed as a binary classification task, where the goal is to predict whether a bank is "dead" (license revoked) or "alive" based on historical financial indicators. To address class imbalance, we match each failed bank with a similar surviving bank using a matching algorithm. We then transform the financial data into image representations which will be used as the input to the CNN model. Our analysis shows that CNNs exhibit high predictive power, particularly when utilizing balance sheet variables. Additionaly, it is seen that the CNN model consistently outperforms the models seen in Rozsos (2022) for longer time spans. The findings confirm that CNNs are an effective tool for bank failure prediction, particularly effective in capturing long-term risk patterns across financial institutions.

Contents

1 Introduction **3**

2 Methodology **5**

2.1 Datasets 5

2.2 Variables 6

2.3 Matching 7

2.3.1 Data Preparation 8

2.3.2 Dead Banks 8

2.3.3 Matching of Alive Banks 10

2.4 Creation of Images 13

2.4.1 Images in Computer Vision 13

2.4.2 Adding Predictors 13

2.5 Variable Sets 14

2.6 Convolutional Neural Networks 15

2.7 Model Specification 16

2.8 Training and Testing 19

3 Results **20**

4 Discussion **22**

4.1 Predictor Importance 22

4.2 Comparison of Results 23

4.3 Limitations and Further Work 24

5 Conclusion **25**

A Training and Validation Accuracies **30**

1 Introduction

Bank failures can have profound and far-reaching consequences, particularly in economies with complex and evolving banking sectors like Russia's. The collapse of even a single bank can trigger systemic risk, where the failure of one institution undermines confidence in others, leading to a chain reaction of financial disruptions (Montagna et al., 2020). This loss of confidence can result in capital flight, disruptions in the payment system, and widespread economic instability, as other banks may face liquidity problems or a loss of market trust. A well-documented example of this cascading effect is the 2008 financial crisis, where the failure of Lehman Brothers caused a global credit freeze, highlighting how interconnected and vulnerable modern financial systems are to single points of failure (Surowiecki, 2009).

In light of the significant consequences that these events have on the economy, understanding and predicting bank failures is crucial to mitigating risk and maintaining economic stability. To this end, many prediction models have been built that tackle the idea of predicting bank failure. These models are essential tools for regulators, investors, and financial institutions to assess the stability of individual banks and the entire banking system.

Most problems can be specified as a binary classification problem where a bank is to be predicted as having failed or still being operational. The model is given data usually in the form of panel data where a comparison is made with multiple banks over a given time period.

Liu et al. (2021) provide a synthesis of the literature on the prediction of bank failures offering insights into the evolution of predictive models and their applications. Early methods for bank failure prediction relied on statistical techniques such as discriminant analysis. Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) were commonly used to classify banks based on financial ratios (Altman, 1968; Sinkey, 1975). As the need for more flexibility arose, logit and probit regression models became widely used for their ability to accommodate binary outcomes and include macroeconomic variables (Martin, 1977; Demirgüç-Kunt & Detragiache, 1998; Arena, 2008). While these traditional statistical methods provided a strong foundation, their performance is subject to multiple limitations and assumptions (Ohlson, 1980). Logit models are sensitive to outliers and probit models are heavily reliant on the assumption of normality.

As computational power grew and more complex datasets became available, researchers began to move beyond traditional linear models, which often struggled to capture the complex and nonlinear relationships present in financial data. In response to the limitations of linear analysis, such as assumptions of normality and independence, neural network approaches gained traction in the early 1990s as a novel alternative for bankruptcy prediction, capable of uncovering subtle patterns in financial data without relying on restrictive assumptions (De Miguel et al., 1993). Studies have shown that Artificial Neural Networks (ANNs) can outperform traditional models in bank failure prediction (Tam, 1991; Bell, 1997; López-Iturriaga et al., 2010).

Prediction using machine learning models like Support Vector Machines (SVMs) (Ecer, 2013) and k-nearest neighbors (KNN) (Le & Viviani, 2018) have also demonstrated high accuracy. Ensemble methods such as Random Forests, AdaBoost, and Gradient Boosting have proven particularly effective due to their robustness and predictive power (Tanaka et al., 2016; Carmona et al., 2019; Kolari et al., 2019; Rozsos, 2022). Models like AdaBoost combine multiple weak learners to form a strong classifier, improving generalization and reducing overfitting (Schapire, n.d.).

In recent years, Convolutional Neural Networks (CNNs) have gained attention for their ability to capture non-linear relationships in data. While these models are traditionally designed for visual inputs, their core strength lies in detecting local patterns through shared convolutional filters that scan across the input. When structured financial data such as a time-series of financial indicators is organized into a grid-like format, these filters can effectively uncover meaningful patterns across both grouped variables (spatial patterns) and time spans (temporal changes). CNNs have shown promise in analyzing irregular trends and abrupt changes in time-series data that traditional linear models may fail to capture (Peng & Yan, 2021). Studies suggest that CNNs can outperform Long-Short Term Memory models (LSTMs), a type of recurrent neural network that is also used for time-series analysis (Wibawa et al., 2022). Other researchers, such as Arratia and Sepúlveda (2020), have also worked with image based approaches, using financial reports transformed into images to improve prediction accuracy. In Chen et al. (2024) we see the use of Temporal Convolutional Networks (Bai et al., 2018) where they use a crisis variable as their target variable, which is a dummy variable that is set to be 1 when one country starts a systemic financial crisis at a particular year.

A key reference for us is Hosaka (2019), who explores the use of CNNs for bankruptcy prediction by converting financial ratios into images. Their approach is to transform the financial data into grayscale images, with each ratio corresponding to a pixel whose brightness reflects the magnitude of the ratio. To optimize the model's learning, correlated financial ratios are positioned close to one another rather than randomly, with placement determined by minimizing an objective function to ensure related pixels remain adjacent.

Our project aims to investigate whether Convolutional Neural Networks (CNNs), which are typically applied in image and spatial data processing, can be adapted for predicting bank failures using structured numeric panel data and whether they will be able to outperform conventional classification models. We will skip the use of the correlation-based optimization used by Hosaka (2019) for image layout and instead adopt a simplified approach by using a pre-arranged dataset where predictors are logically grouped together. For instance, all asset related variables will be on the left side of the matrix. This preserves semantic structure while also reducing the amount of preprocessing needed, allowing the CNN to use its pattern recognition capabilities on financial inputs.

To conclude this section, it is still important to keep in mind that deep learning models, including CNNs, are still faced with the disadvantages of data quality, model interpretability, and computational demands which affect their practical implementations (Dimitrios Gounopoulos et al., 2024).

2 Methodology

2.1 Datasets

The data being used is derived from 2 different data sets. The first dataset is called **panel data** from Karas and Vernikov (2019). This data set contains information on whether or not a bank has failed or been closed. Our target variable is a variable called *revok* which shows when a bank has had it's license revoked. Banks which have had their license revoked will be called **dead banks**. For banks still in operation today, the *revok* variable contains a null value, indicating that the charter has not been revoked. These banks will be referred to as **alive banks**. Each observation would be uniquely identified by a *regn* variable which can be read as an identifier and a date variable *dt* which is usually the first of a month.The predictors would then be derived from the second dataset called **cbrdata**, which contains data available from the website of the Central Bank of Russia (CBR) at https://www.cbr.ru/eng/. The two datasets have varying timelines of data with the **panel** data having data from 1991 to 2020 and the **cbr** data having data only from 2004 to 2020. The latter dataset is incredibly sparse with many of the 919 variables showing null values. On merging we are left with data from January 2004 to September 2020. In total there are observations of 1272 banks.

2.2 Variables

The Central Bank dataset can be seen as having two main parts. A balance sheet and a profit and loss statement.

The balance sheet consists of variables that begin with the letter A and is collected monthly. These variables give us information about a banks financial position at a certain time. The three main components are assets, liabilities and reserves.

1. Assets : Assets are variables that begin with the letter A and contain an a as their 4th character. These represent what the bank owns, including loans and securities. Risky assets can lead to losses, while stable ones ensure steady returns.

2. Liabilities : Liabilities are variables that begin with the letter A and contain an l as their 4th character. These include deposits from customers and other financial obligations. These responsibilities are to be met by the bank and are therefore seen as liabilities.

3. Reserves : Reserves are variables that begin with the letter A and contain a c as their 4th character. These are funds that banks hold to meet regulatory requirements and provide liquidity. They include cash in vaults and deposits at the central bank, ensuring banks can handle withdrawals and financial shocks.

The profit and loss statement of a bank summarizes its revenues, expenses, and profits or losses over a specified period. In the case of our data it is quarterly. To convert this to monthly data and de-cumulate the values we use the method presented in Rozsos (2022). They divide each quarterly value by 3 and use that value for all three months of the quarter.

The two main components of interest are revenues and expenses.

1. Revenues : Revenues are variables that begin with the letter F and contain a r as their 4th character. Revenue is the total income a business earns from its primary operations, like sales of goods or services.

2. Expenses : Liabilities are variables that begin with the letter F and contain an e as their 4th character. Expenses are the costs a bank incurs to operate and generate revenue. A bank's expenses mainly consist of interest expenses, operating costs, and regulatory costs.

We then check if the banks have uploaded the balance sheet and the profit and loss statement. This is done by a variable named *naA* for balance sheets and a variable named *naF* for profit and loss statements. If at least one value has been entered in the balance sheet then we assume it has been submitted and the other values were left empty on purpose. These values are then converted to 0 in the final dataset. The same is then done with the profit and loss statement values. The two variables then hold a boolean value of **True** or **False** which signifies whether the given observation of bank month has a missing statement or not. A value of **True** indicates that the bank has not uploaded the statement for that given month.

2.3 Matching

The merged dataset includes sparse information on 850 dead banks and 422 alive banks across 16 years of data (2004 - 2020). This imbalance in the distribution of our classes can severely impact performance.

We find 6 banks (*regns* : **698**, **1793**, **1816**, **2552**, **3076**, **3371**) appearing in both classes. This occurs because, despite having a *revok* value indicating their date of date, these banks have at least one additional month of data beyond that date, with a non-null *revok* value. As a result, they are mistakenly included among the alive banks as well.

To prevent these problems caused by imbalance and duplication, we aim to create a balanced dataset by going through dead banks and matching each of them with a surviving bank that has data over a comparable time period. This section outlines the process of constructing such a matched and balanced dataset.

2.3.1 Data Preparation

To begin constructing a balanced dataset, we first restrict our data to banks that provide t months of continuous information across a selected set of financial indicators. For the purposes of explanation in this methodology, we set $t = 12$, thereby focusing on banks with a full year of data available prior to death.

The financial indicators selected for this are the Extended CAMEL variables, as outlined in Rozsos (2022), a paper which uses the same dataset and problem of classification. These variables build on earlier work by Kolari et al. (2002), Peresetsky (2011), and Peresetsky et al. (2011) and have more of a focus on ratios to assets that are used to define risk rather than the banks default parameters. The Extended CAMEL variables serve both to filter banks with sufficient data on risk and to match dead banks with comparable banks that are considered alive at the time of their death. These variables are detailed in Tables 1 and 2.

Table 1: Base CAMEL variables

	Variable
1	Log(Assets)
2	Capital / Assets
3	Non-performing loans / Assets
4	Profits / Assets
5	Liquid assets / Assets

2.3.2 Dead Banks

We begin with the pool of dead banks. A bank is considered dead if it has a non-null *revok* value, which marks the revocation of its license. For each dead bank, we collect data from the $t = 12$

Table 2: Extended CAMEL variables

	Variable
1	Capital assets and other non-working assets
2	Non-working assets
3	Non-working assets / Assets
4	Sum of key asset accounts / Assets (quantity squared as in a HHI index)
5	Government bonds
6	Government securities / Assets
7	Non-government securities
8	Non-government securities / Assets
9	Total securities / Assets
10	Loans to non-financial institutions
11	Loans to non-financial institutions / Assets
12	Interbank loans / Assets
13	Loans to households / Assets
14	Total loans and leases / Assets
15	Overdue loans (over 5 days)
16	Overdue loans of firms / Assets
17	Private customers' deposits and accounts
18	Households' deposits / Assets
19	Firms' deposits / Assets
20	Bank reserves for possible losses
21	Reserves for possible losses on loans to non-financial firms / Loans to non-financial firms
22	Provision for loan losses / Assets
23	Net interest income / Assets
24	Nondeposit liabilities / Total liabilities
25	Amounts owed to credit institutions (credits from other banks)
26	Interest revenue / Assets
‘ 27	Interest expenses / Assets
28	Total expenses / Assets
29	Total revenue / Assets
30	(Interest revenue + Interest expenses) / Assets

months preceding its death, including the month of death itself. Rows are arranged in descending chronological order and numbered meaning that the observation with $n = 1$ corresponds to the month of death, and $n = 12$ corresponds to 12 months prior. This is done in order to facilitate consistent slicing for model input. Rows with any missing values in the Extended CAMEL variables are removed, and any bank that lacks 12 full months of clean data is excluded. This ensures we have clean data of exactly a year before death for each bank. This is important as models like CNNs require consistent inputs with images of the same size.

After applying this filter, we identify 581 dead banks for which a complete year ($t = 12$) of data is available. The earliest recorded death here occurred on **22 February 2005** (bank `regn`: **698**), while the most recent death was on **24 September 2020** (bank `regn`: **3122**). For instance, bank 3122's dataset consists of a 12-row array, with the first row representing September 2020 and the twelfth row representing August 2021.

Table 3: Number of dead banks containing t months of information on Extended CAMEL variables.

Time Span (t)	**Number of Banks**
3 months	600
6 months	590
12 months	581

2.3.3 Matching of Alive Banks

The matching algorithm goes through each dead bank, and for each, creates a pool of alive banks that were active during the same time period. For a given dead bank, this pool consists of:

Banks with a *revok* value of null, indicating they never died.

Banks that died two or more years after the death of the selected dead bank.

This allows us to include all banks that were alive at the time of the dead bank's failure, including those that may eventually fail but were healthy relative to the chosen dead bank.

For each candidate alive bank, data is then collected in the same twelve month time window as that of the dead bank, ensuring that both banks have information on the same twelve months. As before, the rows are arranged in descending order, with the first row representing the month of the dead bank's death, and the twelfth row corresponding to twelve months prior. Only those alive banks with complete data for the entire twelve month period are retained. From the resulting pool, one bank is randomly selected to match with the dead bank, preventing bias in the matching process and preserving the representativeness of the alive bank population.

Example (Illustrated in Figure 1) Consider the failed bank with `regn`: **3**, which failed on 27 August 2008. Its final 12-month timeline spans from September 2007 to August 2008. A search for alive banks during this same period yields 833 candidates. Here are two of those candidates:

- Bank `regn`: **1**, which has never died, is included with matching data from September 2007 to August 2008.
- Bank `regn`: **3461**, although dead currently, is also included since its death occurred in 2016 which is well beyond the 2-year threshold.

Both banks, along with the remaining 831, have an equal chance of being matched with bank **3** through random selection. This process is repeated for all 581 dead banks. At the end of the matching, for $t = 12$, we have 581 dead banks and 581 alive banks (with respect to those dead banks), with data on a total of 1162 banks as our balanced dataset. This dataset serves as the foundation for the upcoming binary classification task.

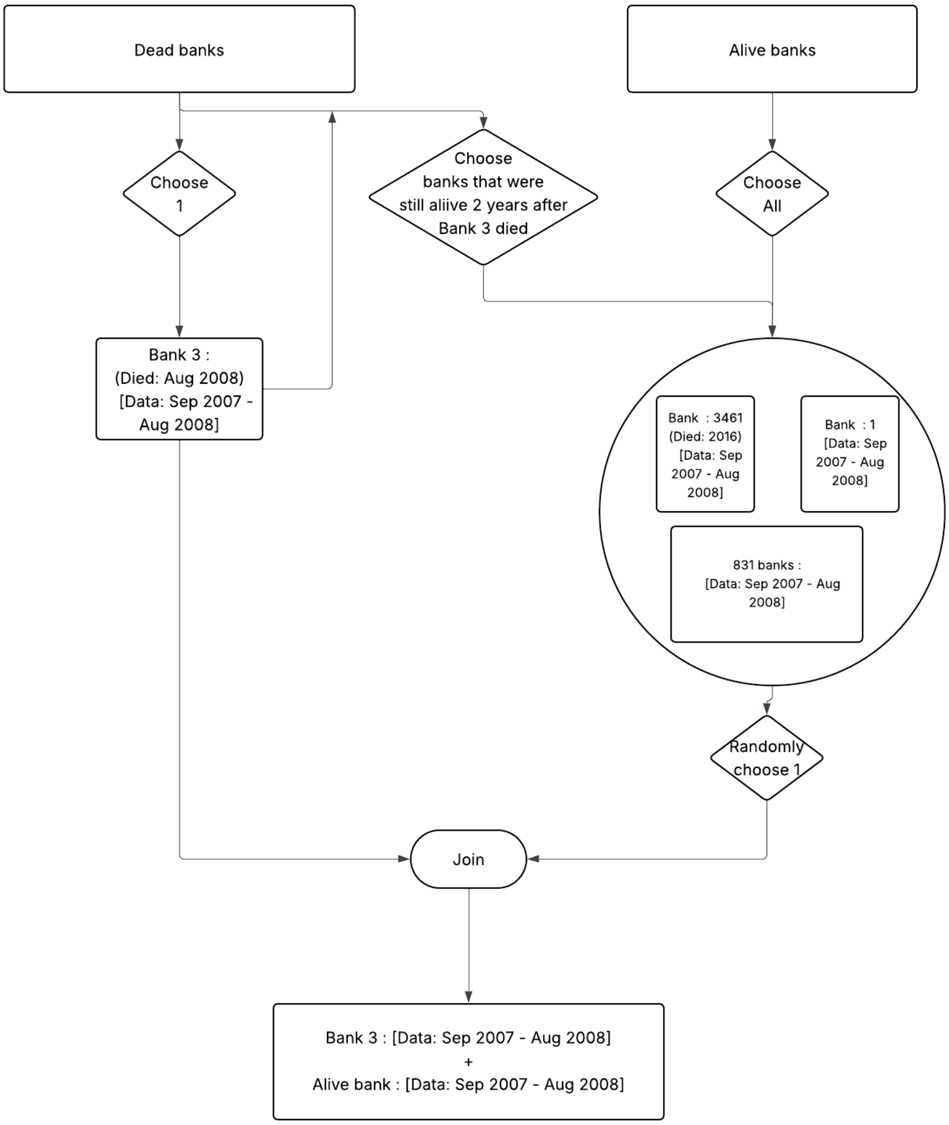

Dead banks
Alive banks
Choose 1
Choose banks that were still aliive 2 years after Bank 3 died
Choose All
Bank 3 : (Died: Aug 2008) [Data: Sep 2007 - Aug 2008]
Bank : 3461 (Died: 2016) [Data: Sep 2007 - Aug 2008]
Bank : 1 [Data: Sep 2007 - Aug 2008]
831 banks : [Data: Sep 2007 - Aug 2008]
Randomly choose 1
Join
Bank 3 : [Data: Sep 2007 - Aug 2008] + Alive bank : [Data: Sep 2007 - Aug 2008]

Figure 1: Example of matching process.

2.4 Creation of Images

2.4.1 Images in Computer Vision

In computer vision, an image is commonly represented as a three-dimensional array (or tensor). The first two dimensions correspond to the image's spatial structure representing its height and width while the third dimension represents the color information through multiple channels. Each position in the height-width grid corresponds to a pixel and in the case of black and white images, each pixel contains a singular value representing an intensity. Therefore, an image of size 900 x 450 would be represented in an array with the shape (900,450,1). Our bank data is also that of a grid like structure where it can be seen as a table or two dimensional array. With one dimension showing the time span, in our case, months and the other dimension being made up of the predictors. Here we still have a third dimension, however it has only one channel, same as the black and white images. The third dimension will be added by reshaping our data. This restructuring of our numeric and tabular data into three dimensional arrays representing black and white images allows the bank data to be used in convolutional neural networks which is the main goal of this project.

2.4.2 Adding Predictors

At the end of the matching we are left with a balanced dataset containing information on dead and alive banks. The predictors for each bank here are the Extended CAMEL variables that were used for matching, these variables are aggregated from raw data. We wish to use un- aggregated raw data as predictors too.

When used in the Extended CAMEL variable set, **assets** refers to one variable, seen in Table 1, which is a summation of 233 variables representing a banks various assets. Here, the **assets** variable is aggregated data and we would want to use the 233 variables, the raw data. This is done by collecting the raw data from the original dataset (before matching) for each bank and corresponding time span from the balanced data set (after matching).

This leads to the array having a shape of (12 x 233). Once the array is created it contains many null values which are converted to zero. We then perform row - wise normalization on the arrays so that all values are scaled between 0 and 1. This approach removes the influence of scale, which is important when working with banks of vastly different sizes, ensuring that comparisons focus on the internal distribution of variables rather than total amounts. Furthermore, in contexts like image analysis or time-series modeling, where each row represents a snapshot in time, the normalizing ensures each image "row" (month) is scaled comparably which allows for the highlighting of changes in the composition over time rather than shifts in overall value.

These arrays are then reshaped to be three dimensional with the third dimension containing 1 channel which finally leads to the creation of an image that can be used as an input for the convolutional neural networks. The final image array when using assets would now have a shape of (12,233,1) where each pixel would represent the normalized value of an asset at a given month.

2.5 Variable Sets

We use different sets of predictors (variables) in the images to see which variables sets offer the best performance in terms of binary classification. Different parts of the balance sheet and profit and loss statement are used, mainly the variables mentioned in section 2.2. Here are the sets of predictors used in the experiments.

1. Balance Sheet and Profit and Loss statement : This includes the assets, liabilities, reserves, revenues and expenses of the bank and amounts to 689 variables. The matching and image creation process results in the formation of images with the dimensions (12,689,1).

2. Balance Sheet: This includes the assets, liabilities and reserves of the bank and amounts to 410 variables. The matching and image creation process results in the formation of images with the dimensions (12,410,1).

3. Profit and Loss statement: This includes only the revenues and expenses of the bank and amounts to 279 variables. The matching and image creation process results in the formation of images with the dimensions (12,279,1).

4. Reserves : This includes liabilities and reserves and amounts to 313 variables. The matching and image creation process results in the formation of an image with the dimensions

(12,313,1).

5. Assets : This includes only assets of the banks and amounts to 233 variables. The matching and image creation process results in the formation of images with the dimensions (12,233,1).

6. Extended CAMELs : This includes all the 35 variables used in the matching processes. The matching and image creation process results in the formation of images with the dimensions (12,35,1).

By organizing the experiments using progressively narrower sets of predictors, from a comprehensive financial overview to specific financial components such as only assets or reserves, the resulting structure enables us to have a clear understanding of which types of financial information are most relevant for classification performance. This also allows for the isolation of where the important information comes from, for example, determining whether liabilities or assets alone contain enough signal for accurate predictions. A strong model using just assets or reserves would indicate minimal input data suffices whereas a model needing full financial information (689 variables) suggests more un - aggregated data is required for classification. The use of the balance sheet and profit and loss statement separately allows us to determine which side contributes to learning and performance more. Including the Extended CAMEL variables in the experiment helps us compare the performance of commonly used aggregated financial indicators with the more detailed, un - aggregated data we are using. It also serves as a baseline, since Rozsos (2022) used this same set of variables in their study.

2.6 Convolutional Neural Networks

Convolutional networks (LeCun, 1989), commonly known as convolutional neural networks (CNNs), are a type of feed forward neural network specifically designed to handle data with a grid-like structure. This includes time-series data, which can be viewed as a one-dimensional grid of values recorded at regular time intervals, and image data, which forms a two-dimensional grid of pixels (Goodfellow et al., 2016). As noted by Li et al. (2021), unlike traditional machine learning methods that rely on manual feature extraction, CNNs can automatically learn relevant features during training, enabling them to capture complex patterns in the data that may not be outwardly apparent. This makes them especially useful for data like images and time - series. CNNs differ from traditional deep learning models by using convolutional layers instead of fully connected (dense) layers to automatically extract spatial or temporal patterns from structured data like images or time series (Chollet, 2018). These layers use filters, also known as kernels, that scan over the image or data grid scanning small regions of the input one at a time . This allows the model to learn localized patterns i.e. spatially dependent relationships where the proximity of features can reveal meaningful information.

We aim to take advantage of the structured layout of our dataset, where certain financial indicators consistently appear in specific regions. For instance, the balance sheet is always positioned to the left of the profit and loss statement and related items like assets and liabilities are placed closer together than more distant categories like assets and expenses. We also intend to skip feature extraction and use un - aggregated data from which the CNN will automatically extract relevant features through the generation of feature maps. The convolution operation produces these feature maps by detecting the presence of specific patterns at various positions. Each element in a feature map reflects the strength with which a particular pattern was detected by the corresponding filter at that location.

Each successive layer outputs more feature maps where increasingly complex information is stored (Chollet, 2018). Once the final convolutional layer has extracted the relevant features, the resulting multi-dimensional tensor is flattened into a one-dimensional vector. This vector is then passed through a fully connected (dense) layer, which serves to combine the extracted features and learn decision boundaries. Finally, the output of this dense layer is fed into a single neuron with a sigmoid activation function, producing a probability score that indicates the predicted class, either “dead” or “alive”.

2.7 Model Specification

The model begins with an input layer that accepts data shaped according to the variable set being used where different sets result in different sizes of images. This is followed by three successive two

- dimensional convolutional layers with increasing numbers of kernels 32, 64, and 128 respectively. Each convolutional layer applies a square kernel of dimension (3 x 3) with 'same' padding, preserving the spatial dimensions of the input at each stage. If we are using images of size (12 x 233) all feature maps in every layer would have the same size. For instance, the first layer would result in 32 feature maps, all of size (12 x 233). The ReLU activation function is used in each convolutional layer to introduce non-linearity and help the model learn complex patterns. After the final convolutional layer, the feature maps are flattened into a one-dimensional vector which is passed to a dense (fully connected) layer with 256 units and ReLU activation. The final layer consists of a single neuron with a sigmoid activation function, outputting a probability between 0 and 1. This value represents the model's confidence in assigning the input to one of two classes: "dead" or "alive". The model is compiled with the Adam optimizer, using a small learning rate (7e-7) and the binary cross-entropy loss function, appropriate for binary classification tasks. Accuracy is used as the evaluation metric.

During experimentation, we began with a standard CNN architecture comprising convolutional layers, max pooling layers and dropout layers. Our initial focus was on tuning the learning rate, and we found that smaller learning rates helped prevent instability during training, which was an issue when larger learning rates caused volatile weight updates and erratic performance. However, this adjustment had limited impact on testing accuracy. To investigate further, we simplified the model by removing pooling and dropout layers, retaining only convolutional layers. This shallow architecture significantly outperformed the original model on testing data, a result which was unexpected, as CNNs typically benefit from pooling and dropout when applied to image data. However, our dataset does not consist of images; instead, it contains structured inputs where each "pixel" (i.e., each element in the input array) represents significant and independent information. In image data, spatial redundancy allows pooling layers to downsample without significant loss, as adjacent pixels often represent the same feature (e.g., part of a cat's ear). In contrast, downsampling in structured data like financial or banking records can eliminate critical information. This characteristic of the data likely explains why the model performed better and is the reason for the unconventional architecture of the final CNN model.

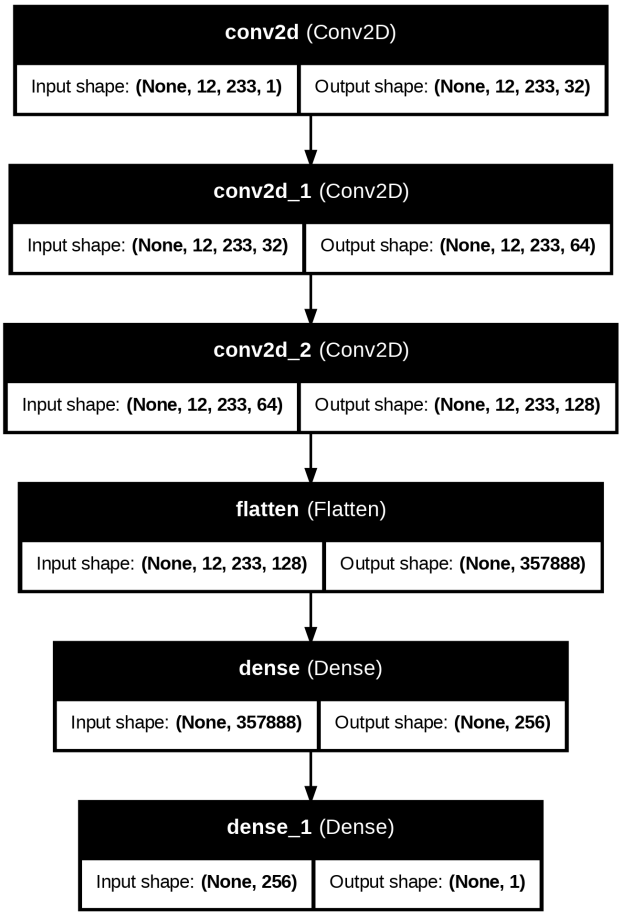
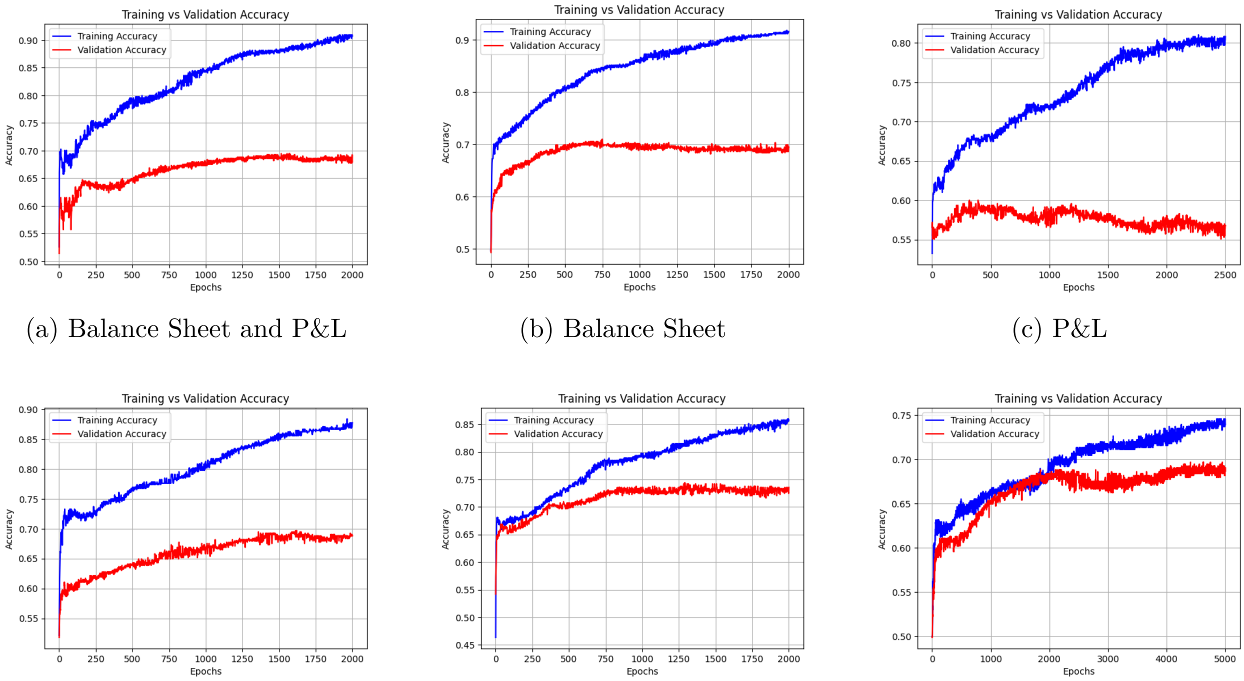

conv2d (Conv2D)
Input shape: (None, 12, 233, 1)
Output shape: (None, 12, 233, 32)
conv2d_1 (Conv2D)
Input shape: (None, 12, 233, 32)
Output shape: (None, 12, 233, 64)
conv2d_2 (Conv2D)
Input shape: (None, 12, 233, 64)
Output shape: (None, 12, 233, 128)
flatten (Flatten)
Input shape: (None, 12, 233, 128)
Output shape: (None, 357888)
dense (Dense)
Input shape: (None, 357888)
Output shape: (None, 256)
dense_1 (Dense)
Input shape: (None, 256)
Output shape: (None, 1)

Figure 2: Sequential Style CNN architecture (sizes of input and output with respect to time span $t = 12$ and use of Assets variable set seen in 2.5).

2.8 Training and Testing

Out of the 1,162 available bank data arrays, 20% were reserved for testing. From the remaining 80% used for training, 50% was further set aside as validation data during the initial training phase. This validation set was used to monitor the model's performance and identify the point at which overfitting began. Across most variable configurations, the model tended to begin overfitting only after a substantial number of epochs, typically not before 750 epochs. After the first training phase for each variable set, the epoch where each one began overfitting can be seen in table 4. Once the epoch corresponding to the peak validation accuracy was identified, the model was retrained from scratch using the full training set (i.e., without holding out a validation subset) until this epoch.



(a) Balance Sheet and P&L

(b) Balance Sheet

(c) P&L

(d) Reserves

(e) Assets

(f) Extended CAMEL

Figure 3: Training and validation accuracies for the first phase of training across different variable sets at $t = 12$. (*Please refer to Appendix A for other time spans.*)

This model was then evaluated on the test set, which comprised 20% of the total data.The performance metrics used were loss, accuracy, and the ROC–AUC score. The loss, specifically binary cross-entropy, was used to measure how well the predicted probabilities matched the true class

Table 4: Number of Epochs Used for Each Variable Set at Different Time Spans

Variable Set	**Epochs** ($t = 3$)	**Epochs** ($t = 6$)	**Epochs** ($t = 12$)
Balance Sheet and P&L	2500	1600	1500
Balance Sheet	2000	2100	750
Profit and Loss	2500	1900	300
Reserves	1250	2400	1600
Assets	750	2500	1200
Extended CAMEL variables	2500	2500	5000

labels. It provides insight into the model's confidence and calibration, even when predictions are correct, low loss values indicate that the model is confidently correct, while high loss may signal uncertainty or poor probability estimates. Accuracy provides a straightforward measure of overall correctness by calculating the proportion of correctly predicted instances. However, to complement these metrics, the Receiver Operating Characteristic – Area Under Curve (ROC–AUC) score was also used. The ROC–AUC score evaluates the model's ability to distinguish between the two classes across all classification thresholds. AUC stands for "Area Under the Curve," and it quantifies the overall ability of the classifier to rank positive instances higher than negative ones. A score of 1.0 represents perfect separation, while a score of 0.5 indicates no discriminative ability (random guessing).

3 Results

The results from testing are seen in Tables 5,6 and 7 along with Figures 4a and 4b.

For $t = 3$, we found that the model using the Balance Sheet and Profit and Loss statement variables had the best performance, with the lowest loss (0.4771), the highest accuracy (80.83%), and the highest ROC–AUC score (0.8570). When using $t = 6$ it was seen that the assets variable set, the smallest of the un - aggregated variable sets had the highest accuracy (80.93%) and with $t = 12$ we found the model using Balance Sheet variables had the best performance, with the lowest loss (0.49), a high accuracy (76.82%), and the highest ROC–AUC score (0.8475). The highest accuracy across all time spans is shown by the model using the assets variables (80.93%) at $t = 12$.

The model using the Extended CAMEL set consistently underperformed compared to most un -

aggregated sets, suggesting worse performance. This model showed it's highest accuracy (72.50%) at $t = 3$. The Profit and Loss variable set showed the worst performance across all time spans showing a peak in accuracy (60.59%) at $t = 6$.

Table 5: Model Performance Across Different Variable Sets for $t = 3$.

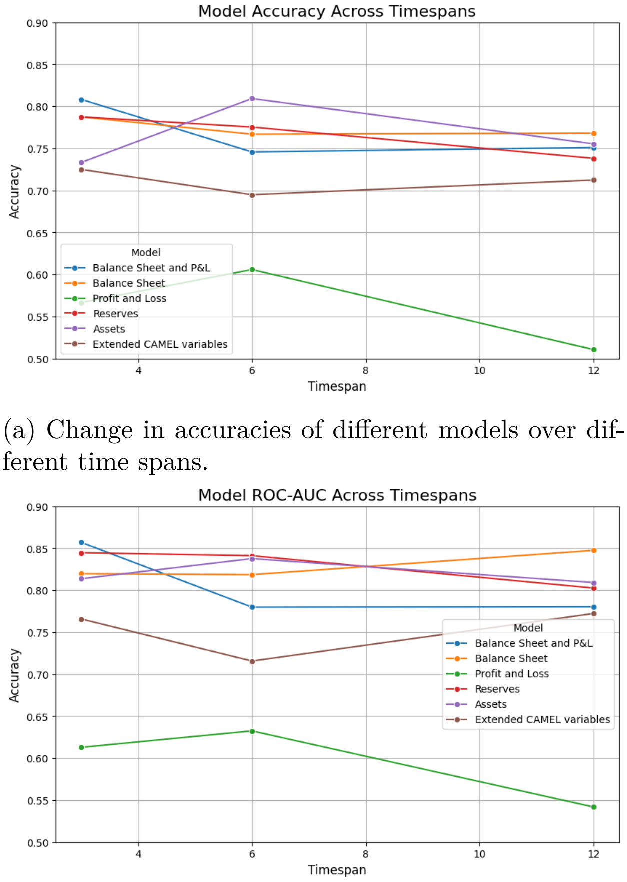
Variable Set	**Loss**	**Accuracy**	**ROC–AUC**
Balance Sheet and P&L	0.4771	0.8083	0.8570
Balance Sheet	0.5353	0.7875	0.8197
Profit and Loss	0.8687	0.5667	0.6129
Reserves	0.4932	0.7875	0.8447
Assets	0.5331	0.7333	0.8137
Extended CAMEL variables	0.5890	0.7250	0.7658

Table 6: Model Performance Across Different Variable Sets for $t = 6$.

Variable Set	**Loss**	**Accuracy**	**ROC–AUC**
Balance Sheet and P&L	0.6062	0.7458	0.7799
Balance Sheet	0.5439	0.7669	0.8186
Profit and Loss	0.7000	0.6059	0.6325
Reserves	0.5326	0.7754	0.8412
Assets	0.5163	0.8093	0.8377
Extended CAMEL variables	0.6221	0.6949	0.7157

Table 7: Model Performance Across Different Variable Sets for $t = 12$.

Variable Set	**Loss**	**Accuracy**	**ROC–AUC**
Balance Sheet and P&L	0.65	0.7511	0.7803
Balance Sheet	0.49	0.7682	0.8475
Profit and Loss	0.71	0.5107	0.5419
Reserves	0.57	0.7382	0.8026
Assets	0.56	0.7554	0.8092
Extended CAMEL variables	0.57	0.7124	0.7723



(a) Change in accuracies of different models over different time spans.

(b) Change in ROC-AUC scores of different models over different time span.

Figure 4: Graphing of results.

4 Discussion

4.1 Predictor Importance

We first observed that most models required many epochs to reach peak validation performance before overfitting, likely due to the absence of pooling or dropout. The absence of dowsampling caused an increase in model capacity where the first dense layer had $\sim$ 91 million parameters. This high capacity reduced the model's ability to generalize (Goodfellow et al., 2016), resulting in slow improvements in validation accuracy.

It was seen that the model using Extended CAMEL variables performed worse than most non - aggregated data models, thereby showing that the CNN worked better with larger and sparse data rather than highly aggregated data of just 35 variables. This could be due to the fact the aggregation did not consider the localized relationships between the variables which lead to a loss of spatial information. We also saw that this model took the longest time or was among the longest to reach a peak validation accuracy, training for around 5000 epochs at $t = 12$, which is very long in contrast to the model using the balance sheet and profit and loss variable set where peak validation accuracy was achieved in just 750 epochs along with a higher testing accuracy (75.54%) for the same time span ($t = 12$).

The worst performing model across all time spans was the one using the profit and loss variable set which showed 56% accuracy at its lowest, suggesting that these variables may not hold much valuable information in regards to a banks health. Most models using various balance sheet variables (especially assets), performed significantly well in every metric across all time spans thereby showing that they contain very significant information on the health of a bank with respect to it's survival. The model using all balance sheet variables was consistently among the best models for all time spans and showed the best performance for time span $t = 12$. Balance sheets are shown to hold more information on risk that the banks takes on (de Haan et al., 2020) and as seen in Kapan and Minoiu (2013), stronger balance sheets are key to credit recovery following a crisis. The consistently high performance of the models using the assets variable sets, seen in larger time spans, shows that these variables are the most important part of the balance sheet in terms of predicting bank failure. Assets like loans and securities represent the bank's core risk bearing activities where their defaulting could incur major losses that are directly absorbed by the bank.

4.2 Comparison of Results

For all three time spans, we find that three or more models show higher accuracies than the best performing model using Extended CAMELS seen in Rozsos (2022) for the same time span.

For $t = 3$, the models using the Reserves (78.75%), Balance sheet (78.75%) and Balance sheet and P&L (80.83%) variable sets outperform the model with the highest accuracy (76.5%) from Rozsos (2022).

For $t = 6$, the models using the Reserves (77.54%), Balance sheet (76.69%), Balance sheet and

P&L (74.58%) and Assets (80.93%) variable sets outperform the model with the highest accuracy (72.5%) from Rozsos (2022).

For $t = 12$, it is seen that all models except the one using the Profit and Loss statement variable set perform better than the model with the highest accuracy (69.4%).

However, this comparison is limited by not being one to one as all of the un - aggregated models do not use the same exact set of 35 Extended CAMEL predictors (Tables 1 and 2) as the models from Rozsos (2022). When using the same predictors as seen in the Extended CAMEL set, the CNN model is shown to have a improvement in accuracy over the best model from Rozsos (2022) (69.4% to 71.2%) only at time span $t = 12$.

In terms of ROC-AUC scores, the sets using the balance sheet variables showed higher scores than the best models from Rozsos (2022) at $t = 6$ and at $t = 12$ indicating a better ability to distinguish between the dead and alive banks over longer periods of time.

In our comparisons with Rozsos(2022), it is seen more and more variable sets outperform their models as the time span increases with cases like model performance at $t = 12$ where five out of the six variable sets show a higher accuracy than the best model from Rozsos (2022). This is especially clear as seen with the highest performing sets at $t = 6$ (Assets) and at $t = 12$ (Balance sheet).

We may conclude that in the context of predicting bank failures, CNN architecture may be better suited to capturing long-term patterns in financial health and provide an advantage over neural networks and machine learning models like Random Forests at larger time spans for prediction. This may be because risk signals tend to repeat over extended periods (Song & Li, 2023), and CNNs are particularly effective at detecting repeating patterns across different parts of a time series. In our case different rows of the image.

4.3 Limitations and Further Work

This analysis was mainly focused on a single failure type and did not go into the heterogeneity of failure types (Rozsos, 2022).

The analysis was exploratory in nature with significant effort devoted to data preprocessing and image generation. This led to two significant limitations, there was no systematic analysis provided for the choice of variable sets and the model building process was not optimized. The former limitation allows us to make comparisons between the different predictors but does not allow for any conclusions on direction of effects or size of effects in general. The latter limitation means that the model is quite shallow in nature with only 5 hidden layers.

The requirement that banks have atleast t months of available data with no null values in any of the 35 Extended CAMEL predictors significantly reduced the amount of data we had available.

We recommend future research to investigate the use of more convolution layers in the model or the use of a hybrid model like the one Cheng et al. (2025) use to improve model performance. A systematic analysis of which predictors to use could benefit model comparison. The use of regulatory ratios along with CAMEL and Extended CAMEL variables in matching and prediction could be used to improve model performance as this set of predictors showed the best results in almost every failure type as seen in Rozsos (2022).

5 Conclusion

This project investigated whether Convolutional Neural Networks (CNNs), typically used with image data, can effectively structure numeric panel data collected from Russian banks. Unlike previous studies which rely on extensive data manipulation (Hosaka, 2019; Arratia and Sep´ulveda, 2020), our approach was able to obtain results using a pre-arranged dataset with minimal preprocessing. When compared to Rozsos (2022), which used the same base dataset and a similar experimental setup with conventional machine learning models and neural networks, our CNN model outperformed those approaches more consistently in longer time spans of analysis. The CNN model demonstrated high predictive accuracy using only asset related variables and achieved strong results on raw, unaggregated data. These findings suggest CNNs offer a novel and promising alternative for bank failure prediction using structured financial data.

References

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, *23*, 589–609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

Arena, M. (2008). Bank failures and bank fundamentals: A comparative analysis of latin america and east asia during the nineties using bank-level data. *Journal of Banking & Finance*, *32*, 299–310. https://doi.org/10.1016/j.jbankfin.2007.03.011

Arratia, A., & Sepúlveda, E. (2020). Convolutional neural networks, image recognition and financial time series forecasting. *Lecture Notes in Computer Science*, 60–69. https://doi.org/10.1007/978-3-030-37720-5_5

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1803.01271

Bell, T. B. (1997). Neural nets or the logit model? a comparison of each model's ability to predict commercial bank failures. *International Journal of Intelligent Systems in Accounting, Finance Management*, *6*, 249–264. https://doi.org/10.1002/(sici)1099-1174(199709)6:3⟨249::aid-isaf125⟩3.0.co;2-h

Carmona, P., Climent, F., & Momparler, A. (2019). Predicting failure in the u.s. banking sector: An extreme gradient boosting approach. *International Review of Economics Finance*, *61*(100), 304–323. https://doi.org/10.1016/j.iref.2018.03.008

Chen, S., Huang, Y., & Ge, L. (2024). An early warning system for financial crises: A temporal convolutional network approach. *Technological and Economic Development of Economy*, *30*, 1–24. https://doi.org/10.3846/tede.2024.20555

Cheng, Y., Xu, Z., Chen, Y., Wang, Y., Lin, Z., & Liu, J. (2025). A deep learning framework integrating cnn and bilstm for financial systemic risk analysis and prediction. https://arxiv.org/abs/2502.06847

Chollet, F. (2018). *Deep learning with python*. Manning, Cop.

De Miguel, L., Revilla, E., Rodríguez, J., & Cano, J. (1993). A comparison between statistical and neural network based methods for predicting bankfailures. *Proceedings of the IIIth International Workshop on Artificial Intelligence in Economics and Management, Portland (USA).*

de Haan, J., Fang, Y., & Jing, Z. (2020). Does the risk on banks' balance sheets predict banking crises? new evidence for developing countries. *International Review of Economics Finance*, *68*, 254–268. https://doi.org/10.1016/j.iref.2020.03.013

Demirguc-Kunt, A., & Detragiache, E. (1998). The determinants of banking crises in developing and developed countries. *Staff Papers - International Monetary Fund*, *45*, 81. https://doi.org/10.2307/3867330

Ecer, F. (2013). Comparing the bank failure prediction performance of neural networks and support vector machines: The turkish case. *Economic Research-Ekonomska Istraživanja*, *26*, 81–98. https://doi.org/10.1080/1331677x.2013.11517623

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Gounopoulos, D., Platanakis, E., Wu, H., & Zhang, W. (2024). Time is the witness: Bank failure prediction via a multistage ai model. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4696313

Hosaka, T. (2019). Bankruptcy prediction using imaged financial ratios and convolutional neural networks. *Expert Systems with Applications*, *117*, 287–299. https://doi.org/10.1016/j.eswa.2018.09.039

Kapan, T., & Minoiu, C. (2013). Balance sheet strength and bank lending during the global financial crisis. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2247185

Karas, A., & Vernikov, A. (2019). Russian bank data: Birth and death, location, acquisitions, deposit insurance participation, state and foreign ownership. *Data in Brief*, *27*, 104560. https://doi.org/10.1016/j.dib.2019.104560

Kolari, J., Glennon, D., Shin, H., & Caputo, M. (2002). Predicting large us commercial bank failures. *Journal of Economics and Business*, *54*, 361–387. https://doi.org/10.1016/s0148-6195(02)00089-9

Kolari, J. W., López-Iturriaga, F. J., & Sanz, I. P. (2019). Predicting european bank stress tests: Survival of the fittest. *Global Finance Journal*, *39*, 44–57. https://doi.org/https://doi.org/10.1016/j.gfj.2018.01.015

Le, H. H., & Viviani, J.-L. (2018). Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios. *Research in International Business and Finance*, *44*, 16–25. https://doi.org/10.1016/j.ribaf.2017.07.104

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*, 1–21. https://doi.org/10.1109/tnnls.2021.3084827

Liu, L. X., Liu, S., & Sathye, M. (2021). Predicting bank failures: A synthesis of literature and directions for future research. *Journal of Risk and Financial Management*, *14*, 474. https://doi.org/10.3390/jrfm14100474

López-Iturriaga, F. J., López-de-Foronda, Ó., & Pastor-Sanz, I. (2010). Predicting bankruptcy using neural networks in the current financial crisis: A study of u.s. commercial banks. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1716204

Martin, D. (1977). Early warning of bank failure : A logit regression approach. *Journal of Banking & Finance*, *1*(3), 249–276. https://ideas.repec.org/a/eee/jbfina/v1y1977i3p249-276.html

Montagna, M., Torri, G., & Covi, G. (2020). On the origin of systemic risk. *SSRN Electronic Journal*, *2050*. https://doi.org/10.2139/ssrn.3699369

Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, *18*, 109–131. https://doi.org/10.2307/2490395

Peng, K., & Yan, G. (2021). A survey on deep learning for financial risk prediction. *Quantitative Finance and Economics*, *5*, 716–737. https://doi.org/10.3934/qfe.2021032

Peresetsky, A. A., Karminsky, A. A., & Golovan, S. V. (2011). Probability of default models of russian banks. *Economic Change and Restructuring*, *44*, 297–334. https://doi.org/10.1007/s10644-011-9103-2

Rozsos, T. (2022, May). Developing bank failure prediction models: Exploring the value of failure type heterogeneity.

Schapire, R. E. (n.d.). Explaining adaboost [Accessed: 2025-05-09]. http://rob.schapire.net/papers/explaining-adaboost.pdf

Sinkey, J. F. (1975). A multivariate statistical analysis of the characteristics of problem banks. *The Journal of Finance*, *30*, 21–36. https://doi.org/10.1111/j.1540-6261.1975.tb03158.x

Song, S., & Li, H. (2023). Unveiling early warning signals of systemic risks in banks: A recurrence network-based approach. https://arxiv.org/abs/2310.10283

Surowiecki, J. (2009, March). Did lehman brothers' failure matter? *The New Yorker*. Retrieved April 21, 2025, from https://www.newyorker.com/business/james-surowiecki/did-lehman-brothers-failure-matter

Tam, K. (1991). Neural network models and the prediction of bank bankruptcy. *Omega*, *19*, 429–445. https://doi.org/10.1016/0305-0483(91)90060-7

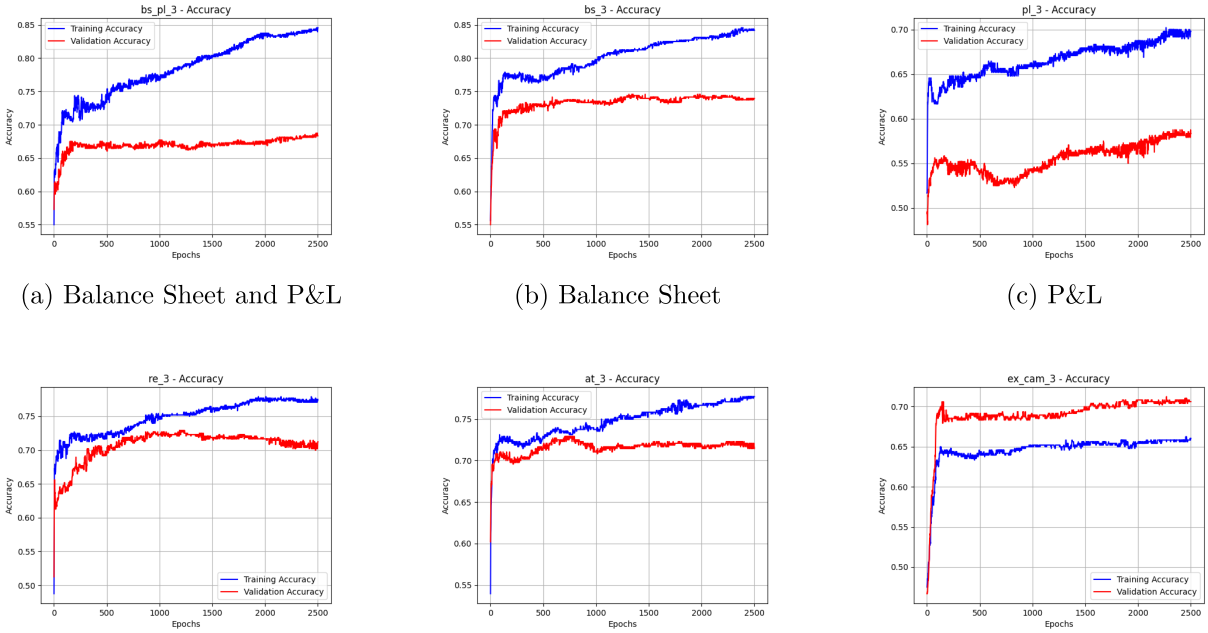
Tanaka, K., Kinkyo, T., & Hamori, S. (2016). Random forests-based early warning system for bank failures. *Economics Letters*, *148*, 118–121. https://doi.org/10.1016/j.econlet.2016.09.024

Wibawa, A. P., Utama, A. B. P., Elmunsyah, H., Pujianto, U., Dwiyanto, F. A., & Hernandez, L. (2022). Time-series analysis with smoothed convolutional neural network. *Journal of Big Data*, *9*. https://doi.org/10.1186/s40537-022-00599-y

Appendices

A Training and Validation Accuracies

Below are visualizations of training and validation accuracy obtained from the initial training phase when using $t = 3$ and $t = 6$.



(a) Balance Sheet and P&L

(b) Balance Sheet

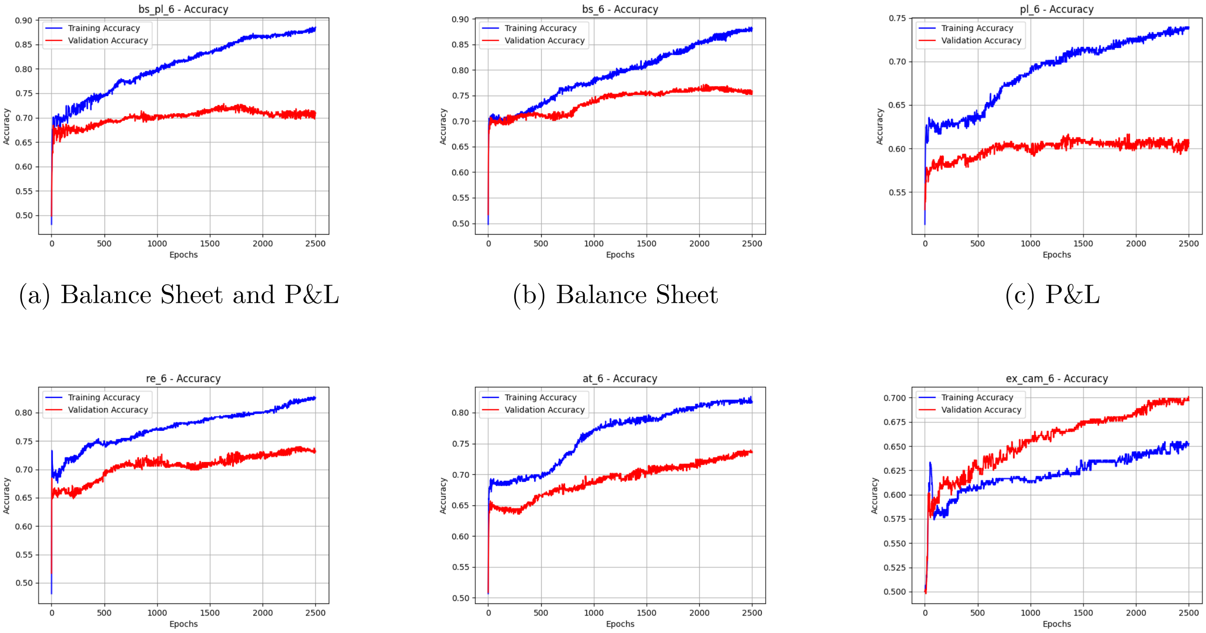
(c) P&L

(d) Reserves

(e) Assets

(f) Extended CAMEL

Figure 5: Training and validation accuracies for the first phase of training across different variable sets at $t = 3$.



(a) Balance Sheet and P&L

(b) Balance Sheet

(c) P&L

(d) Reserves

(e) Assets

(f) Extended CAMEL

Figure 6: Training and validation accuracies for the first phase of training across different variable sets at $t = 6$.