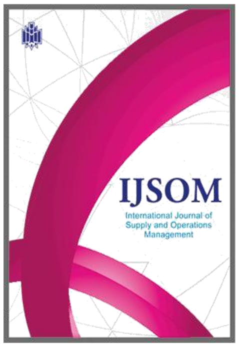
International Journal of Supply and Operations Management

IJSOM

2024, Volume 11, Issue 2, pp. 168-187
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



A Machine Learning Model for Accurate Credit Risk Forecasting in Banking Systems: An Empirical Investigation

Marwa Hasni [a,e*], Mohamed Salah Aguir [b], Mohamed Zied Babai [c] and Zied Jemai [d]

[a] OASIS - ENIT, University of Tunis El Manar, Tunisia
[b] National Engineering school of Carthage, Tunisia
[c] Kedge Business School, France
[a,d] LGI - CentraleSupélec, University of Paris Saclay, France
[e] National Engineering school of Bizert

Abstract

Credit risk consists is the expectation of losses stemming from the inability of a borrower to repay a loan. For the purpose of accurate control of credit risks, banking systems seek developing financial information portfolios upon their customers using sophisticated models which are not only restricted to collecting information on borrower's characteristics, but also, provide visibility on their respective default risk. This paper introduces a novel deep learning model to forecast the credit risk of company customers in banking systems. In particular, we develop a hybrid SVM-LSTM based neural network that predicts the total turnover of a company given the historical data records of its economic and financial features within specific periods. Through an empirical investigation based on data of 13 Tunisian manufacturing and service companies, we show that our proposed model results in more accurate statistical performances compared to the standard LSTM and to the linear regression that is commonly used in the area of credit risk management.

Keywords: Forecasting; Credit Risk Management; Deep Learning; LSTM; SVM; Time Series.

1. Introduction

Credit risk refers to the expectation of losses stemming from the inability of borrowers to repay loans. Inadequate credit risk management has been identified as a significant contributing factor to financial crises, as established in the existing literature. To accurately control credit risk, banks develop financial information portfolios on their customers using comprehensive models which are not only restricted to collecting information on borrowers characteristics, but also, anticipates their respective default risk.

While various models exist in credit risk management research, they can be broadly classified into three groups. The first group comprises theoretical models that criticize conceptual aspects of credit risk, such as probability risk (e.g., Abdesslem et al., 2022), bankruptcy risk (e.g., Kanapickiene et al., 2019), and derivatives pricing (e.g., Su et al., 2021). The second group focuses on linear regression and time series modeling to predict credit defaults, incorporating macroeconomic indicators or a combination of macroeconomic and financial factors (e.g., Hasni and Layeb, 2017).

*Corresponding author email address: Marwa.Gharbi@enit.rnu.tn

DOI: 10.22034/IJSOM.2023.109898.2722

The third group involves multivariate methods, employing classification techniques and discriminant analysis to evaluate credit default scores of companies (e.g., Pang et al., 2021).
Machine learning models have gained considerable attention in credit risk prediction due to their ability to approximate nonlinear and complex patterns using available data. Artificial Neural Networks (ANNs, e.g., Bhatore et al., 2020), k-nearest Neighbor Classifier (e.g., Lin et al., 2021), classification and regression tree (e.g., Tian et al., 2020), case-based reasoning (e.g., Li et al., 2022, Qi et al., 2022), and Support Vector Machines (SVM, e.g., Li et al., 2019; Xia et al., 2021) are distribution-free and well-suited for capturing non-linear relationships. Among them, feedforward architectures have gained prominence due to their flexibility, empirical suitability, and freedom from theoretical assumptions.

However, there exist a number of inconsistencies in these algorithms. Indeed, Backpropagation algorithms which are commonly used in training process, might be slow to converge to the minimum of the error function and can get trapped in local minima. Machine learning are also prone to overfitting, where the model performs well on training data but struggles to generalize to test data. The sequential nature of borrowers' behavioral information is often overlooked by existing models, with limited attempts to handle it.

To address these limitations, this study proposes the integration of Recurrent Neural Networks (RNNs), specifically the Long-Short-Term Memory (LSTM) variant, into credit risk prediction. This allows for less information loss throughout data observations that could be laid over long periods. Moreover, using the LSTM, the network parameters are learnt using its gating mechanism. Henceforth, the aforementioned drawbacks of the usual learning algorithms would be avoided.

Despite their impressive performance, deep learning frameworks can yield ineffective results under certain conditions. To explain the relevance, adequate fitting parameters that are retrieved during the train phase may not generalize well to the test subset. This issue is referred to as overfitting, where the parameter values are too closely tailored to the training subset. To address this problem, two research approaches have been proposed.

The first approach suggests implementing preventive measures, such as dropout operations or adding regularization layers to the network. However, it is important to note that these solutions often require arbitrary parameter choices and lack theoretical foundations.

The second approach advocates developing hybrid deep learning methods to enhance the forecasting task. Available attempts under this heading graft a classification model to the neural network architecture in order to identify homogeneous input sets of data.

In our study, we align with this research strand and attempt to develop a hybrid LSTM architecture whose performance is powered by feeding it with homogeneous subsets of data. In doing this, we got inspired by the cognitive reasoning of human brain that is explained in Festinger and Carlsmith (1959) according to which human brain develops “similar” associations, which will then be compared against a new situation in order to retrieve interpretations for it. Thereby, we propose a new neural network architecture that starts by classifying input data into homogeneous groups before feeding them to the LSTM. In doing this, a classification model is grafted to the LSTM in order to identify homogeneous sets of borrowers.

Selecting a classifier was a challenging task given the available plethora of models. Relevant examples include parametric methods (e.g. Linear Discriminant Analysis: LDA), multi-criteria analysis (e.g. Principal Component Analysis: PCA) and machine learning tools (e.g. SVM, random forest). So, to step into it, threefold criteria have been considered. In particular, the selected classifier should not neither be too time consuming for the purpose of not implying a heavy computation load to the whole model nor require restrictive assumptions on statistical properties to be verified to help to handle variable data patterns. Lastly, corroborative evidence on the performance of the selected model should be proved. Following the first criterion, both parametric and multi-criteria analysis have been dismissed as they typically imply prior statistical tests (e.g. correlation between variables, heteroscedasticity of residuals) to be verified. Hence, focus was directed toward machine learning tools.

Reviewing the relevant literature, we pointed out the significant performance yield upon the appliance of the SVM for solving classification tasks in various fields (Seong-Uk et al., 2021) most notably: intermittent demand forecasts for identifying demand occurrences (e.g. Jiang et al., 2021), gene expression analysis for diagnosing risk disease (e.g.

Alfian et al., 2022), face recognition (e.g. Chaabaneet al., 2022), fault prediction in power systems (e.g. Jian et al. 2019) and credit risk assessment (e.g. Jiang et al., 2018).

Given this corroborative evidence on the effectiveness of the SVM classifier for handling several research questions, we selected it to shape our hybrid LSTM model. As such, we graft the SVM at the input layer of our LSTM architecture to feed it with input information that is clustered into homogeneous groups. In this paper, we demonstrate the way we developed our so-called SVM-LSTM hybrid model and investigate its performance in providing banking systems with accurate information upon their credit market.

The reminder of this paper is organized as follows: The second section delves into the literature dealing with the design and usage of hybrid LSTM models to help to argue our choice of the SVM as an empowering tool for the LSTM and its appropriateness to our research question. The third section quotes the machine learning methods being evolved in our proposed model, including the LSTM and the SVM. We also present the recurrent neural network (RNN) as it is the apparent approach to the LSTM. Detailed description of our SVM-LSTM model is provided in the fourth section. Section 5 outlines experimental settings and validation of the deep learning models included in our study, whereas section 6 reports a comprehensive ***comparative study*** including our proposed deep learning models and the multiple linear regression as a benchmark method. The last section summarizes our research and opens paths for new research questions.

2. Research Background

Our literature review consists of two distinct phases. In the first phase, we examine various approaches for credit risk management and provide a rationale for selecting the LSTM deep learning approach for our study. In the second phase, focus is directed toward different LSTM-based architectures. Through an analysis of their characteristics, advantages, and limitations, we elaborate on the specific hybrid LSTM architecture proposed in our research. This comprehensive review enables us to establish a strong foundation for our study's approach and design.

2.1. Literature Review on the Available Machine Learning Methods for Credit Risk Management

Relevant traditional Methods to deal with credit risk management consist of Linear Discriminant Analysis (L DA) and Logistic Regression (LR). Despite observed performances in establishing visibility within risky context, a number of studies outlined that that these methods are threatened to perform poorly if underlying assumptions are not satisfied. Indeed, Mihalovic (2016) and Gumy and Gumy (2013) have separately investigated shortages underpinning the traditional methods for bankruptcy prediction and pointed out that if particular assumptions such as the independence of explanatory variables or the normality of the data distribution are not satisfied, biased estimates are likely to be obtained.

Substantially, several machine learning techniques have attracted a number of researchers as alternative solution approaches for credit risk management (Shen et al., 2020). With reference to the relevant literature, the available machine learning based contributions may be quoted under three headings. First, we found studies dealing with simple machine learning methods, namely ANN (e.g. Huang et al., 2018), SVM (e.g. Yue et al., 2019), and Classification and Regression Trees (CART) (e.g. Sun et al., 2018). These models have been essentially explored to classify borrowers' status (i.e., default or not) and hence assist decision makers whether a loan may be attributed or not.

Despite observed performances over traditional methods, the existing evidence is unable to assert a performance of one particular model across all case studies. This may be explained by the fact that, machine learning techniques typically require large amount of data and adequate optimization techniques for their parameters. Alternatively, a number of research studies conveyed toward hybrid and ensemble learning techniques which combine multiple machine learning models to help to retrieve benefits from each one of them. Relevant examples include the study in Plawiak et al. (2019) whereby a novel ensemble learning architecture including a cascade of SVM classifiers is proposed. A key feature of this model is that it incorporates evolutionary computation, normalization and feature selection techniques to accomplish effective binary classification of accepted or rejected borrowers. The so-called Deep genetic cascade ensemble of SVM classifiers (DGCEC) has been tested using an extensive data base supplied from an Australian bank and shown to be accurate for about 97% of the cases. As yet, the authors claim their model to outperform all existing peers.

An additional interesting study dealing with ensemble learning is that in Kim and Cho (2019) whereby an SVM-based model is designed for default prediction in social lending. In this study, the authors have been concerned with maximizing the amount of training data to feed to the machine learning model. Indeed, while these models require data to be labeled, the latter are difficult to provide for ongoing loans. This yields less training data which impacts the effectiveness of the machine learning model. Alternatively, Kim and Cho (2019) combined label propagation and transudative Support Vector Machine (TSVM) with Dempster-Shafer theory to handle both labelled and unlabeled data. Briefly, label propagation groups data with similar features into the same class, whereas TSVM identifies and separates data with different features. At the outcome, the Dempster-Shafer fusion method combines the outputs of these two techniques to achieve accurate labeling.

This ensemble method has been shown to improve the accuracy and the precision scores of the prediction task when compared against a peer method that has been priorly proposed in Li et al. (2017) for predicting loan applicant status using semi-supervised SVM.

To sum up, usage of machine learning techniques enabled for more accurate predictions of default risk for least costs (Shen et al., 2020). However, in credit scoring applications dealing with large datasets, traditional statistical and machine learning methods have demonstrated serious difficulties to unveil the complex interrelationships among credit data variables. This has motivated researches to investigate the deep learning models. In particular, the Long-Short-Term Memory (LSTM) network, have been widely used for credit risk assessment, most notably, when dealing with big data for credit scoring applications. Zhang et al. (2017) implemented a single LSTM network for credit score evaluation in peer-to-peer lending and obtained more accurate classification when compared against traditional statistical and machine learning techniques. A quiet similar study may also be quoted in Wang et al. (2018) whereby a so-called attention LSTM mechanism has been applied as a consumer credit scoring method based on borrowers' online behavior. Interested readers to supplementary studies dealing with deep learning for credit risk management could find good reference in Sezer et al. (2020) whereby a comprehensive review on relevant deep learning architectures are reported. This study also emphasizes the outperformance of deep learning architecture over their traditional machine learning peers.

At the outcome, a number of key aspects upon deep learning techniques, machine learning and traditional statistical methods could be pointed out: To start with, deep learning and machine learning techniques enabled for more accurate credit risk mitigation, which helps banks and financial institutions avoid financial losses. Their ability to incorporate specific techniques such as feature selection and data normalization has enabled for lower credit analysis cost (e.g. Kim and Cho, 2019). Using machine learning models, faster loan decisions are taken given their ability to process large amounts of data quickly. Lastly, machine learning models and most notably deep learning showcased a superior performance in handling complex credit risk systems such as those involving big data and complex relationships between variables.

Given this evidence, we have been motivated by implementing a deep learning based model for the purpose of forecasting credit risk. As we are dealing with time series, we firstly selected the LSTM as a base classifier since it has demonstrated wide empirical evidence in the related literature (Sezer et al., 2020).

2.2. Literature Review on Available Hybrid LSTM Models for Credit Risk Management

While LSTM networks showcased effectiveness in capturing long-term dependencies in sequential data, they may still face limitations in certain scenarios.

One major limitation consists in the fact that LSTM networks can be computationally intensive, most notably when dealing with large-scale datasets or complex architectures (Pascanu et al., 2013, Vu et al., 2023). As such, since significant computational resources and time are required, LSTM might reveal less adequate for resource-constrained contexts.

Furthermore, While LSTM networks are designed to capture long-term dependencies, they may struggle with certain types of dependencies that span very long sequences. In particular, if the relevant information is spread too far apart in the sequence, LSTMs may have difficulty retaining and utilizing that information effectively. Lin et al. (2017) addressed this issue by incorporating a structured self-attention mechanism that assists the model to reach relevant parts of the sequence. Here, we shall point out that this novel solution has been applied in the context of natural language processing which cannot be adapted to our numerical time series data.

At the outcome, it could be stated that these challenges triggered academics for developing improvement solutions. As previously mentioned, there are two research approaches that have been proposed, whereby the first suggests the implementation of preventive measures, such as dropout operations or the addition of regularization layers to the network. However, it should be noted that these solutions often necessitate arbitrary parameter choices and lack solid theoretical foundations. Ultimately optimization processes such as the GridSearch cross validation (GridSearchCV) could intentionally be explored to help to identify the best combination of parameters that fits the data the best. Nevertheless, this procedure depends from the initially proposed benchmark values of parameters.

On the question of the second approach, it advocates the development of hybrid deep learning methods that aim to improve the forecasting task. In this context, several attempts have been made to integrate a classification model into the neural network architecture. The purpose of this integration is to identify homogeneous sets of input data, allowing for more accurate and effective analysis.

In our work, focus was directed toward this second research strand. Henceforth, two choices could be taken either to implement an initial layer with a read-out function (i.e. a classification function typically used in the final layer to filter input data) or else, implement a whole classifier and graft it at the input of the main forecasting model.

The available literature is tightened up upon the improvement of the ANN architectures that deal with classification problems by challenging their conventionally used read-out function: the softmax regression. To explain the relevance, substantial outputs h_t, undergo the softmax activation function which converts them into probability actions used to identify their respective classes over possible ones. As may be noticed, the softmax is a probabilistic approach that requires a high computational cost. This motivated a number of researchers to replace the softmax by distribution-free classifiers to help to avoid training the model under the constraint of building a particular probability distribution and hence allows for faster computations that could handle variable data as lesser iterations shall be encountered.

Amongst, support vector machine (hereafter SVM) has revealed of a promising performance, most notably for binary-classification. Alalshekmubarak and Smith (2013) developed a model using SVM for the output layer of the Echo State Network (i.e. a variant of RNN) and showed its outperformance over its standard version that uses the softmax, of about 2% in terms of accuracy. In addition, when dealing with reduced data samples, the ESN-SVM achieved higher accuracy than ESN of about 15%. Thus far, it could be stated that SVM empowers the accuracy of ANN patterns. This has motivated us to explore it in our proposed model. Yet, SVM are found to achieve satisfactory results when applied to economic and financial data (Tang et al., 2022). In this vein, we quote the study in Altan et al. (2019) which used SVM to predict the closing prices of USD/TRY and EUR/TRY exchange rates given financial time series data, including daily closing price, minimum price, maximum price, and trading volume parameters. A key aspect of this study is that it assessed the impact of various kernel scale values, key parameter for the SVM approximation, and selected the most valuable one for financial time series forecasts.

Additional hybrid LSTM models include the so-called DBN-LSTM network which combines LSTM networks with a deep belief network (DBN), whereby, the DBN is used to refine the features being fed to the LSTM. This model has been designed to particularly deal with stock price prediction and has compared more favorable than traditional method (i.e. ARIMA) as well as traditional machine learning tools (i.e. LSTM, RNN and Multi-Layer perceptron). A quiet similar study in Bao et al. (2017) proposes a hybrid model that incorporates stacked autoencoders and LSTM networks for financial time series prediction. The stacked autoencoders are used for extracting relevant features prior to process the data into the LSTM. Interested readers to more studies dealing with the development of hybrid LSTM in the area of financial predictions could find good references in Htun et al. (2023) a comprehensive survey on the usage of specific models for empowering the LSTM is provided.

Except the study in Alalshekmubarak and Smith (2013), all reviewed studies dealing with hybrid LSTM have been concerned with refining the features being introduced to the neural network architecture.

Veritably, this yields an inevitable information loss since a filter is priorly applied, which might not be suitable for deep learning models that typically require a big amount of data to ensure accurate performances. Meanwhile, we shall recall that we are interested in modeling the aforementioned cognitive reasoning by analogies, which relies upon likewise data.

As such, we propose to graft the SVM at the input of our selected LSTM model to feed it with information that is refined into homogeneous groups rather than reduced.
Our choice of the SVM classifier is underpinned by its corroborative empirical evidence for ensuring accurate classification even for non-extensive datasets. In addition, this method is less greedy in terms of validation hypothesis to ensure, prior to explore it. Indeed, this machine learning tool is able to handle both linearly and non-linearly separable data. A relevant study that emphasizes these properties is that in Liu et al. (2013) whereby an extensive comparative investigation is carried out, including SVM, the random forest machine learning classifier and Back propagation neural network to classify electronic tongue data for the recognition of orange beverage and Chinese vinegar. This study showcases the outperformance of the SVM in terms of three classification metrics, namely: recall, precision and the F1-score.
Recently, Kurani et al. (2013) conducted a comparative study to assess the performance of the SVM and Artificial Neural Network (ANN) for stock prediction. In this research, the main focus was to address the overfitting issue, which occurs when a model becomes too tailored to the training data to an extend that it fails to generalize when dealing with unseen data. Throughout this study, the authors have been able to put evidence on the effectiveness of the SVM model in overcoming the overfitting issue when compared to the ANN.

3. Theory and Methods

In this section, we give theory and technical details underpinning the machine learning methods evolved in our study.

3.1. Deep-Learning Based Methods

The contribution of this paper mainly relies upon the Long Short-Term Memory (LSTM) which is a deep learning architecture. From a technical point of view, the LSTM is an improved variant to a previous architecture called the Recurrent neural network (hereafter RNN). Given this technical linkage, we propose explaining this apparent method prior to outline that of the LSTM.

3.1.1. Recurent Neural Network (RNN)

Applications of ANNs have revealed prominent in handling patterns recognition and classification tasks. However, their performance is found to be questionable when dealing with sequential data. Indeed, while predictions should be built on the basis of previously gained knowledge, traditional ANNs do not involve ways to conserve it across underlying layers. To address this shortage, a substantial neural network architecture, labelled recurrent neural network (RNN) has been developed under the Markov assumption that each state depends only on the last state. RNNs encompass three layers, namely: input, hidden and output layers that are stacked by means of loops that allow information to persist. Just like what does a traditional network, input layers receive incoming stimuli to produce an output information. The difference is that neurons are fed with new information that is combined with its previous state. Going into more details, consider figure 1 whereby information h_{t-1} is concatenated to the new inputs x_t. Afterwards, activation function f is applied elementwise to their respective weighted versions according to the equation (1):

$$f(W_{hx}.x_t + W_{hh}.h_{t-1}). \qquad (1)$$

This corresponds to a new hidden state h_t used to calculate the neuron output $o_t = W_{hy}.h_t$ and to feed next neuron. Lastly, we note that common choices for activation functions are *tanh* and *ReLu*.

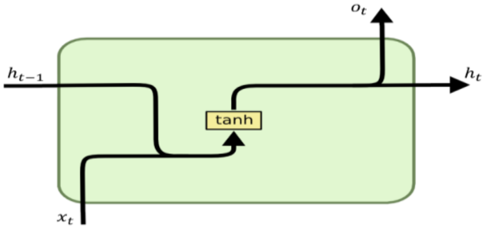


Figure 1. RNN cell

3.1.2. Long Short Term Memory (LSTM)

Though being satisfactory, RNN accuracy has been challenged on the ground that for long time steps it yields an inevitable loss of older information. Indeed, long time steps enhance the number of hidden layers, which are associated with different sets of weights. As such, running the learning process favors the fading of the weights of older layers and hence prevent retrieving information from corresponding neurons. In addition, it should be noted that the RNN cells do not control the amount of information that should be kept forward from that that needs to be forgotten. An additional major drawback consists in the exploding and vanishing problems that might underpin the gradient. Here, we shall note that when training an ANN, the underlying layers undergo a backtracking process with a view to determine the optimal set of weights of neurons that minimize a loss (an error), also referred to as gradient. Accordingly, if the adjusted weights are small (less than 1), the resulting gradient would be minor leading to the vanishing gradient problem. Conversely, larger weights yield very high gradient values that consist in the exploding issue. In short, this phenomenon is referred to as the scaling effect.

These shortages motivated researchers to design a variant to the RNN cell, which is the long short-term memory (LSTM) cell in order to mend them. In particular, to fix the vanishing/exploding gradient threats, a scaling factor set to one is considered and enriched by gating units building the whole LSTM cell. Besides, to accurately handle long-term dependencies, LSTMs provide an output vector presenting the memory cell in addition to the state cell vector. The memory cell is specifically designed to manipulate available data through insertion and/or drop operations that would prevent losing useful information for further analysis tasks (e.g. predictions, classification). Mainly, the LSTM cell operates by updating the memory cell using forget and input gates. Afterwards, the cell memory thus obtained, together with available data information are explored to generate the new cell state.

The structure of an LSTM cell is outlined in figure 2 and uses the following notation:

x_t: The vector of the incoming information at point time t.
h_{t-1}: The vector of the value of the previous hidden layer.
h_t: The vector of the value of the output layer.
C_{t-1}: The vector of the memory cell at point time $t-1$.
C_t: The vector of the memory cell at point time t
$\sigma, tanh$: Activation functions used to filter data
W_{hh}: Matrix based on the Previous Hidden State
W_{hx}: Matrix based on the Current Input
W_{hy}: Matrix based between hidden state and output
b_i: bias

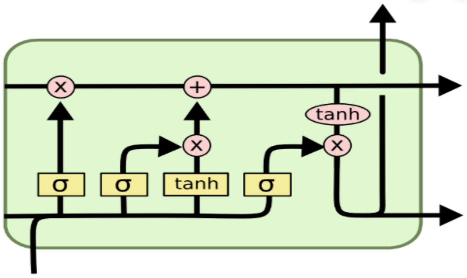


Figure 2. LSTM cell

The following steps display the way according to which it acts.

Step 0: Concatenate x_t and h_{t-1} to obtain an input vector, say v_t.
Phase 1: Update of the cell memory
Step 1: Draw v_t under the sigmoid function elementwise to distinguish between useful and unuseful data for upcoming analysis tasks. This is given by equation (2):

$$f_t = \sigma(W_{hx}.x_t + W_{hh}.h_{t-1} + b_i) \tag{2}$$

Step 2: Decide on the data that should be completely delated from that that should be maintained by evaluating the matrix product in equation (3):

$$C_t = f_t * C_{t-1} \tag{3}$$

Step 3: Propose new candidate inputs to be added to the cell memory using the $tanh$:

$$\check{C}_t = tanh(W'_{hx}.x_t + W'_{hh}.h_{t-1} + b_i) \tag{4}$$

Step 4: Decide on the candidates that should be introduced to the cell memory by evaluating equation (5):

$$i_t * \check{C}_t \tag{5}$$

Whereby:

$$i_t = \sigma(W'_{hx}.x_t + W'_{hh}.h_{t-1} + b_i) \tag{6}$$

Step 5: Add the obtained vector to the previous cell memory to update it:

$$C_t \coloneqq C_t + i_t * \check{C}_t \tag{7}$$

Phase 2: Define the cell state at point time t

Step 6: Select eligible incoming information using equation (8):

$$o_t = \sigma(W''_{hx}.x_t + W''_{hh}.h_{t-1} + b_i) \tag{8}$$

Step 7: Control the memory cell by applying *tanh* according to equation (9)

$$\tanh(C_t) \tag{9}$$

Step 8: Generate the output state by evaluating equation (10):

$$\mathrm{h_t} = \mathrm{o_t} * \tanh(C_t) \tag{10}$$

3.2. Support Vector Machine (SVM)

In statistical learning theory, support vector machine (SVM) is a classification tool whose broad objective is to define the optimal hyperplane that splits data into homogenous groups by maximizing the margin line. SVM is a versatile technique that has been largely developed and explored in a number of machine learning tasks. Relevant examples include pattern recognition (e.g. Chen and Xie, 2007), classification tasks (e.g. Chaudhuri, 2016) as well as regression and time series forecasting (e.g. Weston et al., 1999). The SVM model may be described in the following way:

Consider a subset of the training data points:

$$\vartheta \subset D = \{(x_i, y_i) | x_i \in R^p, y_i \in \{-1, +1\}, i = 1..n\} \tag{11}$$

Estimate the decision function equation of the form:

$$f(x) = \langle w. k_{x_n} \rangle + b \tag{12}$$

With $w \in \vartheta$, k is a kernel function, b is the bias term and $\langle .,. \rangle$ denotes the dot product.

In practice, estimating this function in minimizing the total deviations from real targets y_i over the training set while maintaining its flatness as big as possible. Analytically, this corresponds to the optimization of the equation (13):

$$\mathrm{d(x)} = \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{p} max(0, |\xi| - \varepsilon) \tag{13}$$

Where ξ is a cost function, C is the regularization parameter and $C \sum_{i=1}^{p} max(0, |\xi| - \varepsilon)$ determines the trade-off between the flatness of f and the amount up to which larger deviations than ε are allowed.

4. The Proposed Hybrid SVM-LSTM Model

As priory mentioned, LSTMs are designed to undertake situations where RNN fail, most notably to bridge long time lags and to fix the vanishing/exploding gradients. Additional pros that have been revealed consist in handling variable data and continuous values. Also, with LSTMs, updating weights are O (1) less complex than that with RNNs. However, some cons have also been pointed out as far as LSTMs got explored in a number of fields (e.g. language modelling, image processing, speech and handwriting recognition). Indeed, these nets are threatened to perform poorly if court memory bandwidth is available or if they got fed with very random weights due to data variability. As may be noticed, these shortages are linked to the quality of the input data. In particular, if a high amount of data is available, it is likely to encompass high randomness that would be harmful for LSTMs. Therefore, to tune this noise and enhance the net remembrance, we find it judicious to explore quiet coherent input data. To do so, we propose a neural network architecture that grafts an SVM layer to an LSTM network for the purpose of clustering the data prior to processing them for forecasts. Thereby, the LSTM will be fed by homogeneous data subsets rather than a single heterogeneous sample. By proceeding so, we attempt to improve the learning performance of the model. Yet, it should be pointed out that this additional task would not generate significant computation time, as the SVM does not need to satisfy a probability distribution. As such, not only does the SVM provide rather stable range of classes, but it would reduce the total computation time. Mainly this idea is inspired from the cognitive reasoning by analogies that has been demonstrated in Tam, and Kiang (1992). Figure 3 depicts the architecture of our proposed neural network. To start with, we conduct a prelaminar analysis upon the available data in order to verify the eligibility for a prior classification of data prior to process them for forecasts. As is well known, the SVM is a supervised learning algorithm whereby the number of classes should be pre-specified. However, when dealing with a dataset, one cannot fairly decide on the optimal number of classes that should be used. For that, we referred to the elbow method which is a heuristic that allows to determine the number of homogeneous groups within a dataset. The main principal consists in simulating possible values for the number of clusters and plotting the related explained variation. As such, the elbow of the curve reflects the optimal number of clusters that should be used. In our study, this number is pointed out and fed to the SVM to help to draw homogeneous clusters of data accordingly.

Turning back to our framework, we set up an input layer to receive incoming information, which has been initially converted into a sequence of chronological events. This sequence is then passed onto the SVM classifier to generate likewise subsamples according to the number specified upon the elbow methodology. As is well known, a kernel function for the determination of the optimal hyperplanes that separate the data the best should be selected. Since we are dealing with high-dimensional and overlapping data, we selected the radial basis function (RBF) kernel. Accordingly, data is classified based on the relative squared distances between observations in a Euclidian space. More specifically, the evaluated metric reflects the influence that a particular observation exerts on the other. As such, the further two observations are from each other, the less influence they have on each other and hence they are likely to belong to different classes. The structured data thus obtained is converted into supervised fashion. More specifically, the data is split into features and target values. Here, we shall point out that we are dealing with a multivariate LSTM whereby a number of macroeconomic and financial features are explored to predict future turnover. So, technically, we refer to a many to one LSTM configuration according to which data structure should be prepared in a way to split features from the target value. The data thus obtained is plugged into the LSTM network that is designed by extending four hidden layers emanating from an input layer. As may be noticed from our chart flow, we call a dropout technique upon feeding LSTM with the input data. This technique is recommended upon the training phase to help to add more variability to data. Indeed, the LSTM, upon each training epoch, the LSTM randomly dismiss 20% of the data. As such, the training phase would be exerted upon different subsamples. This would be of interest to help the machine learning learn from versatile data and hence prevents the overfitting issue. The four hidden layers' process input data as outlined in figure 3 and scrap the output to a dense layer that uses a particular activation function to get the predictions of the borrower's turnover. Here, we shall point out that the number of neurons within subsequent hidden layers are decreasing in a way to converge to a single value. To do so, we explore the halving procedure which is an ad-hoc method that favors the convergence of the values toward one single value (Yang et al., 2020)

5. Performance Investigation and Parameter Settings

When dealing with deep learning models, the following cohort is typically followed after implementing the proposed model:

Step 1: Collection and preparation of data.

Step 2: Optimization of the hyper-parameters of the model.

Step 3: Training the model.

Step 4: Exploration tasks (i.e., forecasts and/or classification)

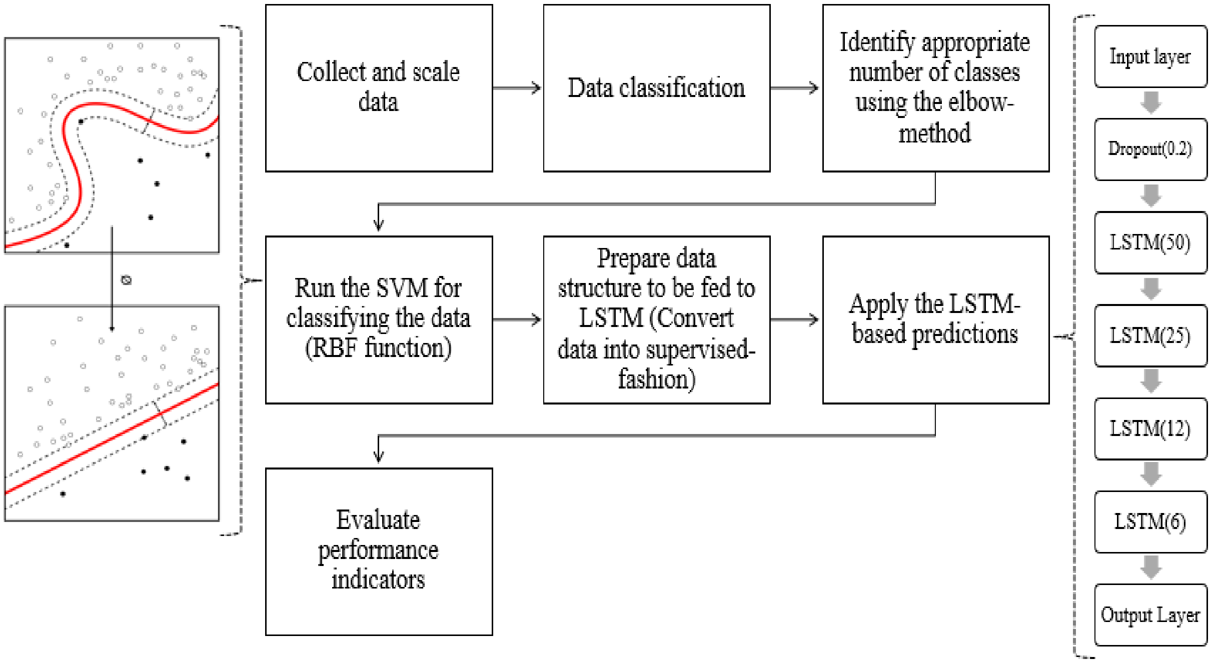

Collect and scale data
Data classification
Identify appropriate number of classes using the elbow-method
Run the SVM for classifying the data (RBF function)
Prepare data structure to be fed to LSTM (Convert data into supervised-fashion)
Apply the LSTM-based predictions
Evaluate performance indicators
Input layer
Dropout(0.2)
LSTM(50)
LSTM(25)
LSTM(12)
LSTM(6)
Output Layer

Figure 3. Framework of the proposed SVM-LSTM hybrid model

5.1. Collecting and Preparing Data

We refer to a private Tunisian bank, which seeks developing portfolios of information on future turnovers of its affiliated companies to help to make the decision whether a loan could be approved or not, if any is requested. In particular, we are interested in modeling the turnovers of 13 Tunisian companies operating in several sectors such as the automobile, equipment distribution and agro alimentary. We engaged in discussions with experts in the considered bank to select eight macroeconomic and financial input variables. Macroeconomic variables consist in Inflation Rate (IR), Unemployment Rate (CR), the Growth Rate (GR) and the External Balance of Trade balance (EBT). Selection of these factors is inspired from the “magic square” in Kaldor (1971), which is a graphical representation linking the four main objectives of conjectural economic policy of a country. In economy studies, it is believed that this representation is successful and it is labeled by the expression of “magic square” to highlight how the simultaneous achievement of the four goals yields the economic performance of the country. Moreover, we notice that numerical data being used are historical records laying from 2006 to 2015 praised by the National Statistical Institution (NSI) in Tunisia. On the question of the financial factors, we engaged in discussions with experts in the case study bank to pick-up four accounting indicators, namely: The Equity Ratio (ER), the Current Asset (CA), the Current Liability (CL), the Net Income (NI), the own capital (OC) and the total Brut Income (BI). Table 1 outlines key statistics describing the macroeconomic and the financial variables for calculating the turnover of a training sample of 200 companies and figure 4 depicts associated boxplots. As may be noticed, most indicators are with rather stable variability except for the EBT, the IR and the CR indicators which show significant fluctuations. This variability is mainly justified by the instable economic situation in Tunisia. With regard to our forecasting models, this dataset is found to be heterogeneous and hence the forecasting task would be challenging as models should handle different aspects.

Table 1. Statistical description of companies' turnovers

	mean	**std**	**min**	**25%**	**50%**	**75%**	**max**
BI	177566	343706	85	45171	70583,5	153836	2370083
OC	62858,2	98521,1	-9781	20785	33255,5	53311,8	590335
CL	79356,9	185526	21	17018,3	29025,5	56739,8	1423564
RN	4659,47	27283	-205284	608	3021,5	7053	112465
CA	79864,4	134815	48	27644,8	42146	70575,8	1042244
GR	3,2	2,27726	-2	3	3,5	4	7
EBT	-6,2	3,467	-11	-9	-6	-3	-2
IR	4,1	1,04665	2	4	4	5	6
CR	14,6	2,20552	12	13	14	16	18
Turnover	154263	247707	28	37258,3	60418	100050	1220071

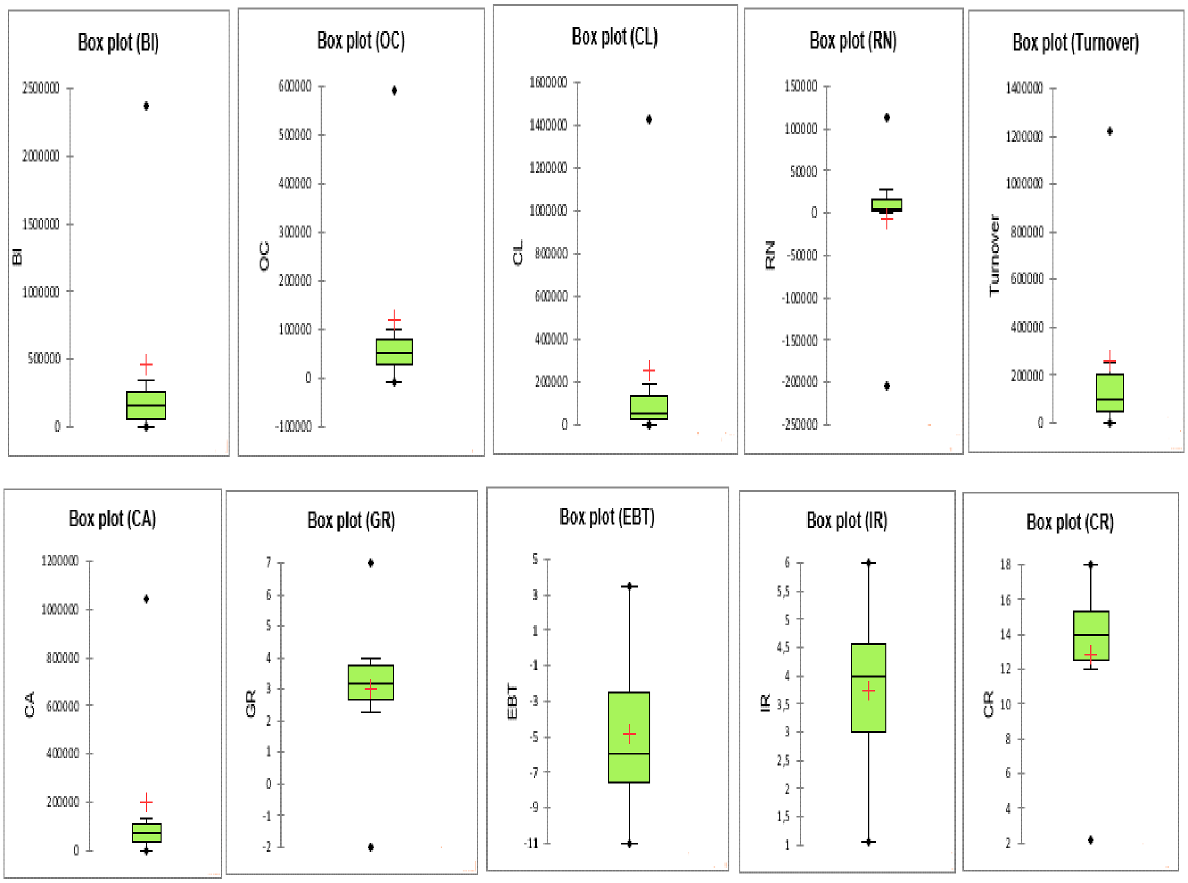


Figure 4. Boxplots of the features and the target

5.2. Optimization of HyperParameters

When dealing with deep learning models, it is key to make use of an adequate framework that encompasses tools for model parameters optimization, training and test tasks. For this purpose, it is common practice to refer to adequate optimization algorithms to help to retrieve the optimal values. This could be achieved using cross-validation. In machine learning literature, a plethora of cross-validation techniques have been designed. Amongst, the K-fold is widely used. This method, known as K-Fold Cross-Validation, partitions the data into K equally sized subsamples, commonly referred to as "folds." Each fold is used as a validation set once, while the remaining K-1 folds are combined as the training set. The process is repeated K times, ensuring that each data point appears exactly once in the validation set. Each training-validation cycle identifies adequate fitting parameters to the particular training subset and then assess their relevance using the validation subset. As such, a performance measurement, which differ according to the nature of the target, is computed. In our study, we used the mean squared error validation metric as the target is a continuous quantitative variable (i.e. turnover rate). Lastly, a comprehensive evaluation of the model's performance is computed by averaging the results from all K validation cycles.

To take full advantage of cross-validation, we combined it with the GridSearchCV algorithm. While cross-validation allows for a more reliable estimation of the model's performance by evaluating the model on multiple training and validation subsets, the Grid SearchCV allows to specify a grid of values for the model's hyperparameters that we want to test. As such, scenarios yield by all possible combinations of these hyperparameters are built and undergo the cross-validation process to evaluate the model's performance for each configuration. Finally, the model with best parameters is retained for the test phase. For the purpose of our study, we used
the Grid search algorithm for optimizing three hyper-parameters, namely the batch size of the minimal-lot (the number of the available sub-samples), the epochs number (the number of times the training process is repeated), and the activation function.

Table 2 depicts the set of tested values whereas the retained ones are bolded. As may be noticed, four batch sizes are tested, namely 50, 100, 150 and 200. When deciding this set of values, we started by the maximal number which is the size of the whole dataset encompassing 200 companies. Then we arbitrary sat a 50-step decreasing values. By proceeding so, we attempt to take into account the impact of data partitioning on the optimal hyperparameters. More specifically small batch sizes like 50 and 100 may result in more frequent weight updates, which help the model converge faster and imply less memory resources. However, they might cause more noise in the weight updates due to the inherent randomness in smaller sample sizes which, in turn, could lead to fluctuations in the training process and make convergence less stable. On the other hand, high batch sizes such as 150 and 200 enable for smoother weight updates and hence, more stable training process. Potentially, they result with faster training. Nevertheless, one should bear in mind that with larger batch sizes, the model is threatened to fail to generalize and to be more prone to overfitting as the risk to converge toward local minima is higher.

When selecting the set for the number of epochs careful attention was directed toward two factors, namely: the complexity of the model and the size of the available data. As previously described, our hybrid LSTM-SVM model can be viewed as complex since it incorporates two machine learning models, amongst a deep learning architecture that encompasses multiple hidden layers. Moreover, we implemented a number of optimization techniques namely: the dropout and regularization to help to optimize the analysis. Taken together, these aspects support strong recommendation to set out longer training times to capture intricate patterns in the data. This range of values is inspired by the Bootstrap methodology which is a sampling methodology commonly used for intermittent demand forecasts (i.e., Hasni et al., 2019) that typically iterates overs 1000 replications on samples for producing one predictive value. With reference to the optimal parameters emanating from the GridSearchCV, we found 2500 and 2000 epochs for the hybrid LSTM-SVM and the standard LSTM, respectively. In our sense, lesser epochs for the standard LSTM are meaningful as it consists of a simpler architecture than the SVM-LSTM.

On the question of the activation function, we assessed two variants, namely the hyperbolic tangent (hereafter tanh) and the Rectified Linear Unit (hereafter reLu). Despite suspicious that reLu is best fitting to our data given the fact that the prediction task deals with continuous quantitative variable, we assessed the performance of tanh which allows to normalize the data and capture non-linearity. However, the GridSearchCV retained the reLu. Indeed, this activation function has gained immense popularity in deep neural networks given a number of advantages, most notably, reLu allows neurons to adapt and activate only when the input is positive. This enables the model to learn more efficiently by focusing on relevant features in the data. Moreover, reLu is shown to be a versatile activation function that is theoretically able to approximate any continuous function. This property highlights the expressive power of reLu-based networks.

To train our models, we selected 75% of the total data points and processed them using the ADAM (adaptive moment estimation) under the RMSE-typed loss function. Table 3 outlines the underlying RMSE-based training performances and shows that our proposed hybrid model is better trained than its standard counterpart is. Thereby it could be asserted that clustering the data empowers the learning performance of the LSTM network.

Table 2. Optimal Hyperparameters of LSTM models

	SVM-LSTM	**Standard LSTM**
Batch size	50-100-150-**200**	50-100-150-**200**
Number of epochs	500-1000-2000-**2500**	500-1000-**2000**-2500
Activation function	tanh, **reLu**	tanh,**reLu**

Table 3. RMSE-based training performances of the LSTM models

Deep learning model	**Train**
LSTM-SVM	17342,26
LSTM	18384,65

Recall from section 4 when introducing our framework, that we attempt to optimize the number of categories to recommend to the SVM classifier. The output of this step is given in figure 5 whereby we plot the variance yield by each proposed number of classes ranging from 2 to 11. As may be noticed, the elbow of the curve is recoded when simulating a 6-category attempt. Henceforth, this value would be used when designing the SVM architecture. This may be perceived as an additional manual parameter optimization that we carried out to furtherly empower our proposed hybrid SVM-LSTM model.

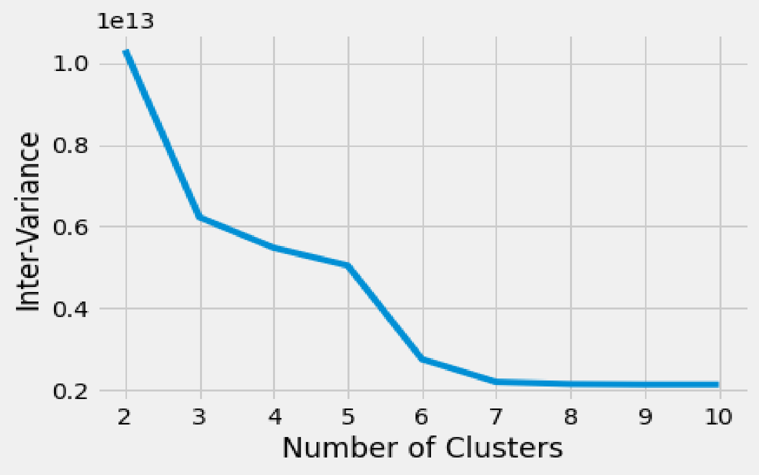


Figure 5. Optimal Number of classes using the elbow-method

6. Comparison of the Forecasting Performances

This section is split into three subsections that progressively report the comparative analysis that have been conducted for the purpose of our study. First, we introduce the evolved models and provide detailed description on the way we implemented and validated the multiple linear regression model. Afterwards, we provide a comprehensive description on the way deep learning models have been tested. Lastly, the overall comparative results are depicted in the third subsection using key performance metrics for credit risk management.

6.1. Involved Models

At this stage of our study we have introduced our hybrid deep learning model that incorporates SVM with LSTM. Afterwards, we tackled the training phase using the K-fold cross-validation technique within the GridSearchCV algorithm. At the outcome, best models of both standard and hybrid SVM-LSTM architectures are obtained. In this section, we propose a comparative investigation that puts in competition these two deep learning models to help to highlight forecasting improvement that is yield by the hybrid model. Here, we wish to point out that including the standard LSTM could also be viewed as a benchmark method as it is widely used in the relevant literature. Related examples include the study in Maleki et al. (2021) and Ala'raj et al. (2021).

To better highlight the performance of the newly proposed model, we enriched our comparative study with the multiple linear regression (LR), which is a benchmark method in the bank,and widely recognized as powerful statistical tool in the early literature dealing with credit risk management.

In particular, we concentrated on designing 13 multiple linear regression models. In doing this, we make use of the stepwise regression approach to select the least number of variables that yield the best coefficient of determination. The following steps summarize our methodology:

Step 1: Write the general linear model using all available exogenous variables.
Step 2: Evaluate the coefficient of determination (R^2) between each exogenous variable and the endogenous counterpart.
Step 3: Dismiss the least significant variables (i.e. $R^2 < 0.70$) and obtain the final general regression model.
Step 4: Estimate the models parameters using the least squares method.
Step 5: Investigate the statistical significance of the model to assert that the dependence between exogenous and endogenous variables is not simply a chance.
Step 6: verify whether the obtained model meets the multiple linear regression hypothesis by:
Step 6.1: Testing the homoscedasticity of residuals.
Step 6.2: Testing the non-correlation between errors

Table 4 highlights statistical tests being conducted in steps 5 and 6 as well as decision rules to be considered for the model validation, whereas table 5 outlines the underlying numerical results which assert the validity of the linear regression model.

Table 4. Statistical tests

Evaluation	**Used statistic**	**Decision rule**
Quality of the adjustment	R^2	Maintain variables whose $R^2 > 0.7$
Statistical significance of the LR model	Fischer-test	The model is statistically significant if P-value <0.02
Normality of residuals	Kolmogorov-Simirnov-test 5% error risk	Residuals are normally distributed if the P-value > error risk
Homoscedasticity of residuals	White-test 5% error risk	Homoscedasticity of residuals is asserted if the P-value >error risk

Table 5. Statistical test results for the validation of the LR models

Company	**Quality of adjustment R^2**	**Fisher test**	**Kolmogorov-Simirnov Test**	**White test**
A	0.984	<0.0001	0.738	0.318
B	0.996	0.015	0.907	1.000
C	1.000	<0.0001	0.587	0.762
D	0.92	0.002	0.583	0.999
E	1.000	<0.0001	0.831	0.968
F	0.924	<0.0001	0.623	0.091
G	1.000	0.0014	0.730	1.000
H	0.992	<0.0001	0.975	0.233
I	1.000	0.000	0.994	1.000
J	0.990	0.002	0.893	0.554
K	0.72	0.002	0.985	0.210
L	0.972	<0.0001	0.904	0.350
M	1.000	<0.0001	0.587	0.762

6.2. Testing the deep learning-based models

In this section, the performances of the two trained deep learning models are evaluated and compared for the purpose of forecasting future turnovers of companies, given their accounting and macroeconomic features. The test subset consists of 25% of the remaining data from the initial provided sample, which corresponds to about 13 companies. Thereby, as the dimension of the test set is not required to be expanded as for the training subset, we carried out the test simulations for each company separately. In our sense, this would yield more comprehensive forecasts of a direct practical relevance for the bank decision makers. However, prior to exert this task, it is important to check that both test and train sets are derived from quiet similar distributions. Otherwise, the training step would be unmeaningful and the machine learning algorithms would perform poorly as they have been trained using different data distribution from that of the test set. To do so, we start by plotting boxplots showing the variations of key statistics of the turnover distributions of each of the 13 companies. These are sketched in figure 6.

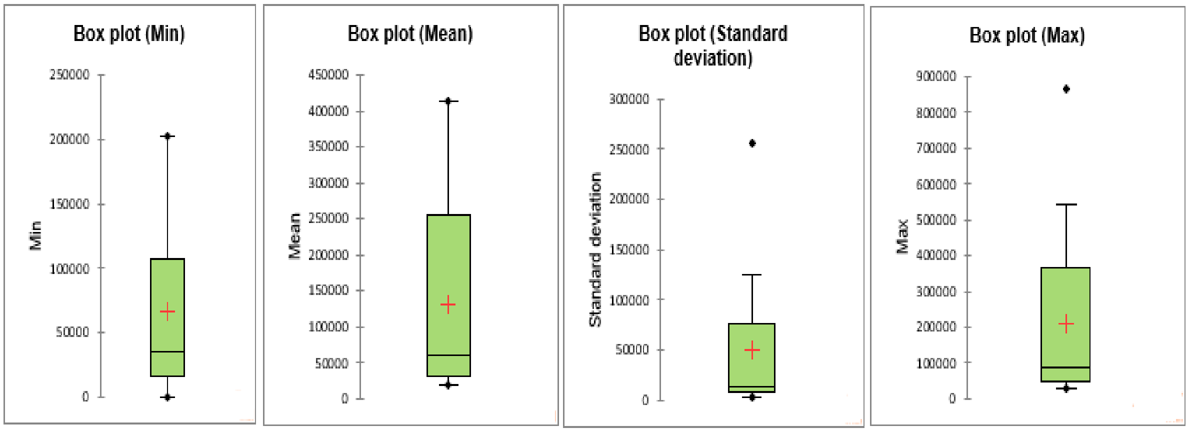


Figure 6. Boxplots of the key position statistics of the turnovers of the test set companies

At the outcome, all tested companies are found to have quiet variable turnover values given their different performances and the distinguished activity sectors they operate in. Yet, a common aspect across all indicators, except for the standard deviation, is that they are right skewed as the majority of the values lay on the right side of their respective mean values. The standard deviation informs on a low fluctuation underpinning the mean turnovers. Throughout this descriptive analysis we put evidence on the overall trend underpinning the variability of the set subset which reveals comparable to that recognized on the train subset.

To further assert that both train and test sets are quite similar we also carry out the Mann–Whitney U test. This is a non-parametric hypothesis test that allows to compare two distributions. The null hypothesis asserts that the data samples emanate from the same distribution under an error-risk. To check this hypothesis, a p-value is calculated and compared against the pre-specified error-risk. If the p-value reveals superior to the risk-error, the null hypothesis could be accepted. For the purpose of our study, we evaluated the U Mann–Whitney statistic to compare the train and the test sets under a 0.05 error risk. The resulting statistic reveals of 713 with a p-value of 0.405. Henceforth, it could be concluded that both test and train sets are similar and might be interpreted as emanating from similar probability distributions. Thereby, we assert that the training of our deep learning approaches is meaningful for the test phase.

6.3. Empirical results and discussion

Tables 6 outlines the RMSE statistics yield by the appliance of each forecasting method. As may be noticed, the LSTM models perform better than the linear regression with a significant preference to the SVM-LSTM variant. This finding goes in accordance with the existing research strand which highlights the substantial benefits emanating from the use of artificial intelligence tools. In addition, it asserts our initial hypothesis that learning from rather stable range could improve the LSTM forecasting performances.

Table 6. RMSE performance comparisons between the SVM-LSTM, LSTM, and LR models

Company	***SVM-LSTM***	***Standard LSTM***	***LR***
A	5442.3	6210.3	17909.3
B	16973	18298.	34297.5
C	2231.1	2614.47	11481.13
D	33577.9	48792.	86927.1
E	1071.3	2205.7	27963.4
F	17899	21297	690522
G	10012.2	14269	20783.85
H	58195.6	90957	302280.7
I	114489.0	139613	228073.1
J	2303.89	3342.3	5701.64
K	5086.66	5655.9	16912.11
L	1605.16	4452	5447.71
M	3534.93	3270	15234.41

To take forward our comparative analysis, we considered two additional performance metrics that are widely recognized in the literature dealing with credit scoring, the area under the curve (hereafter AUC) and the Kolmogorov–Smirnov statistic (hereafter KS) (Shen, 2020). The KS metric is used to assess the predictive power of the models. In credit risk management, the KS statistic is typically used to evaluate the ability of a credit scoring model to differentiate between good and bad borrowers. It measures the maximum difference between the cumulative distributions of predicted probabilities for the positive class (defaults) and the negative class (non-defaults). As such, a classification model showing high KS value is considered as effective in accurately distinguishing between borrowers' classes and ranking them. On the other hand, the AUC indicator.

ROC (Receiver Operating Characteristic) is a scalar value representing the classifier's ability to distinguish between default and non-default cases. Here, we shall point out that this indicator emanates from the well-known graphical tool: Receiver Operating Characteristic (ROC). Thereby, a higher AUC value indicates better discrimination and suggests that the model is better at correctly classifying borrowers as good or bad risks.

Table 7 depicts the average AUC and KS scores for the competing models and their associated standard deviations. Here, we shall notice that the AUC score cannot be evaluated for the linear regression model as it does not provide binary classification. As may be noticed, the hybrid SVM-LSTM model compared more favorable than its standards counterpart and the multiple linear regression model. Furthermore, the hybrid SVM-LSTM outperformed in terms of the AUC indicator. In addition, variation around the observed performances is relatively low (in the order of 2%) which leads to conclude a rather stable performance.

Using both of these indicators along with the aforementioned RMSE lead to develop a comprehensive evaluation of the relative performances of the compared models. The results thus obtained enriches the existing evidence in the literature on the fact that LSTM architectures are of promising performance for credit risk scoring (Worku et al., 2019). In particular, substantial improvements are likely to be obtained when incorporating key machine learning models for specific enhancement purposes.

Table 7. KS and AUC performance comparisons between the SVM-LSTM, LSTM, and LR models

Evaluated model	**Kolmogorov–Smirnov (KS)**	**Area under the curve (AUC)**
SVM-LSTM	86.11%±2.31%	66.67%±2.40%
Standard LSTM	80.56%±2.51%	61.02%±2.10%
LR	79.52%±1.61%	---

7. Conclusions and Perspectives

This study proposes a credit management tool for the practice of a loan officer from a Tunisian commercial bank, using artificial intelligence. Our proposed model is a novel application of deep learning, which grafts an SVM layer to an LSTM network to feed it with stable ranges of data instead of a single heterogeneous set. This yields a hybrid multivariate LSTM network that classifies inputs before exerting the prediction task. By proceeding so, we mimic the human brain way of reasoning by analogy. As such, information is grouped into small and coherent pieces for easy remembrance and faster analysis. The hybrid SVM-LSTM model compared more favorable than its standard counterpart (LSTM) and the benchmark linear regression (LR) analysis when applied for the purpose of forecasting future turnovers of bank borrowers. This has been established in terms of three statistical metrics, namely the RMSE, the KS and the AUCDespite the outperformance of our proposed hybrid model, the gap to ideal frontiers related to the AUC is of about 32%. This leads to question the performance of this model for different datasets most notably highly imbalanced ones. To cope with, we suggest incorporating the Synthetic Minority Over-Sampling Technique (SMOTE) a data augmentation technique that allows to empower the ability of machine learning models to learn from minority class yield by imbalanced data. This technique has empirical corroborative evidence in the area of credit risk forecasting (Shen, 2020) and hence could yield promising results when grafted to our hybrid SVM-LSTM model.

In this study, we only focused on clustering data prior to the forecasting task. Despite making use of the elbow method to retrieve appropriate number of classes, other alternatives for refining input data, such as padding regularization layers should also be investigated. Furthermore, it would be useful to investigate the trade between a particular forecasting horizon and how long the LSTM model should look-back from the available history to retrieve useful information for forecasts.

References

Abdesslem, R. B., Chkir, I., & Dabbou, H. (2022). Is managerial ability a moderator? The effect of credit risk and liquidity risk on the likelihood of bank default. *International Review of Financial Analysis, 80*, 102044.

Alalshekmubarak, A., Smith, L. S. (2013, March). A novel approach combining recurrent neural network and support vector machines for time series classification. In 2013 9th International Conference on Innovations in Information Technology (IIT) (pp. 42-47). IEEE.

Ala'raj, M., Abbod, M. F., & Majdalawieh, M. (2021). Modelling customers credit card behaviour using bidirectional LSTM neural networks. *Journal of Big Data, 8*(1), 1-27.

Alfian, G., Syafrudin, M., Fahrurrozi, I., Fitriyani, N. L., Atmaji, F. T. D., Widodo, T., ... & Rhee, J. (2022). Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method. *Computers, 11*(9), 136.

Altan, A., & Karasu, S. (2019). The effect of kernel values in support vector machine to forecasting performance of financial time series. *The Journal of Cognitive Systems, 4*(1), 17-21.

Babu, N. R., and Mohan, B. J. (2017). Fault classification in power systems using EMD and SVM. *Ain Shams Engineering Journal, 8*(2), 103-111.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one, 12*(7), e0180944.

Barboza, F., Kimura, H., Sobreiro, V. A., Basso, L. F. (2016). Credit risk: from a systematic literature review to future directions. *Corporate Ownership & Control, 13*(3), 326-346.

Bhatore, S., Mohan, L., & Reddy, Y. R. (2020). Machine learning techniques for credit risk evaluation: a systematic literature review. *Journal of Banking and Financial Technology, 4*, 111-138.

Chaabane, S. B., Hijji, M., Harrabi, R., & Seddik, H. (2022). Face recognition based on statistical features and SVM classifier. *Multimedia Tools and Applications, 81*(6), 8767-8784.

Chandra, M. A., and Bedi, S. S. (2021). Survey on SVM and their application in image classification. *International Journal of Information Technology, 13*, 1-11.

Cuentas, S., Garcia, E., & Penabaena-Niebles, R. (2022). An SVM-GA based monitoring system for pattern recognition of autocorrelated processes. *Soft Computing, 26*(11), 5159-5178.

Festinger, L., Carlsmith, J. M. (1959). Cognitive consequences of forced compliance. *The journal of abnormal and social psychology, 58*(2), 203.

Hasni, M. and Bhar Layeb, S. (2017). Multiple regression-based models for accurate credit risk management, International Journal of Economics & Strategic Management of Business Process (Special Issue: The 5th International Conference on Innovation & Engineering - IEM2017), volume 10, Issue 1, pages 53-57.

Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation, 9*(1), 26.

Huang, X., Liu, X., & Ren, Y. (2018). Enterprise credit risk evaluation based on neural network algorithm. *Cognitive Systems Research, 52*, 317-324.

Jiang, J., Chen, R., Chen, M., Wang, W., & Zhang, C. (2019). Dynamic fault prediction of power transformers based on hidden Markov model of dissolved gases analysis. *IEEE Transactions on Power Delivery, 34*(4), 1393-1400.

Jiang, H., Ching, W. K., Yiu, K. F. C., and Qiu, Y. (2018). Stationary Mahalanobis kernel SVM for credit risk evaluation. *Applied Soft Computing, 71*, 407-417.

Jiang, P., Huang, Y., and Liu, X. (2021). Intermittent demand forecasting for spare parts in the heavy-duty vehicle industry: a support vector machine method. *International Journal of Production Research, 59*(24), 7423-7440.

Kaldor, N. (1971). Conflicts in national economic objectives. The Economic Journal, 81(321), 1-16.

Kanapickiene, R., & Spicas, R. (2019). Credit risk assessment model for small and micro-enterprises: The case of Lithuania. *Risks, 7*(2), 67.

Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science, 10*(1), 183-208.

Li, W., Paraschiv, F., & Sermpinis, G. (2022). A data-driven explainable case-based reasoning approach for financial risk detection. *Quantitative Finance, 22*(12), 2257-2274.

Li, Y., Li, Y., and Li, Y. (2019). What factors are influencing credit card customer's default behavior in China? A study based on survival analysis. *Physica A: Statistical Mechanics and its Applications, 526*, 120861.

Lin, J., and Han, L. (2021). Lattice clustering and its application in credit risk management of commercial banks. *Procedia Computer Science, 183*, 145-151.

Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.

Liu, M., Wang, M., Wang, J., & Li, D. (2013). Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. *Sensors and Actuators B: Chemical, 177*, 970-980.

Liu, Q., Tan, T., & Yu, K. (2016, November). An investigation on deep learning with beta stabilizer. In 2016 IEEE 13th International Conference on Signal Processing (ICSP) (pp. 557-561). IEEE.

Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., Gucci, S. (2017). Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics, 183*, 116-128.

Maleki, M. S., Motevallian, S. N., Hosseini, F., Sabokrou, M., & Maleki, H. S. (2021, October). Improvement of credit scoring by lstm autoencoder model. In 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE) (pp. 182-187). IEEE.

Narayan, Y., Ahlawat, V., & Kumar, S. (2020). Pattern recognition of sEMG signals using DWT based feature and SVM Classifier. *International Journal of Advance and Science Technology, 29*(10), 2243-2256.

Pang, S., Hou, X., and Xia, L. (2021). Borrowers' credit quality scoring model and applications, with default discriminant analysis based on the extreme learning machine. *Technological Forecasting and Social Change, 165*, 120462.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2013). How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026.

Qi, P., Wang, F., Huang, Y., and Yang, X. (2022). Integrating functional data analysis with case-based reasoning for hypertension prognosis and diagnosis based on real-world electronic health records. *BMC Medical Informatics and Decision Making, 22*(1), 149.

Seong-Uk Nam, Sangil Kim, HyunMin Kim, and YongBin Yu. (2021). Comparative study of the performance of support vector machines with various kernels. *East Asian Mathematical Journal*, *37*(3), 333-354.

Su, C. W., Cai, X. Y., Qin, M., Tao, R., & Umar, M. (2021). Can bank credit withstand falling house price in China? *International Review of Economics & Finance, 71*, 257-267.

Sun, J., Lang, J., Fujita, H., & Li, H. (2018). Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. *Information Sciences, 425*, 76-91.

Tam, K. Y., Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management science, 38*(7), 926-947.

Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., ... & Li, J. (2022). A survey on machine learning models for financial time series forecasting. *Neurocomputing, 512*, 363-380.

Tian, Z., Xiao, J., Feng, H., & Wei, Y. (2020). Credit risk assessment based on gradient boosting decision tree. *Procedia Computer Science, 174*, 150-160.

Vu, M. T., Jardani, A., Krimissa, M., Zaoui, F., & Massei, N. (2023). Large-scale seasonal forecasts of river discharge by coupling local and global datasets with a stacked neural network: Case for the Loire River system. *Science of The Total Environment*, 165494.

Wang, C., Han, D., Liu, Q., Luo, S. (2018). A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism LSTM. *IEEE Access, 7*, 2161-2168.

Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., & Liu, Q. (2018, February). Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the eleventh ACM international conference on web search and data mining (pp. 592-600).

Worku, Y. B., & Muchie, M. (2019). The Uptake of E-Commerce Services in Johannesburg. *Civil Engineering Journal*, *5*(2), 349-362.

Wu, Y., Xu, Y., & Li, J. (2019). Feature construction for fraudulent credit card cash-out detection. *Decision Support Systems*, *127*, 113155.

Xia, Y., Li, Y., He, L., Xu, Y., & Meng, Y. (2021). Incorporating multilevel macroeconomic variables into credit scoring for online consumer lending. *Electronic Commerce Research and Applications, 49*, 101095.

Xiao, C., Xia, W., & Jiang, J. (2020). Stock price forecast based on combined model of ARI-MA-LS-SVM. *Neural Computing and Applications*, *32*, 5379-5388.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing, 415*, 295-316.