Received: 9 February 2022 | Revised: 14 December 2022 | Accepted: 8 January 2023

DOI: 10.1002/cpe.7637

WILEY

RESEARCH ARTICLE

Intelligent credit scoring using deep learning methods

Adaleta Gicić[1] | Dženana Đonko[1] | Abdulhamit Subasi[2,3]

[1]Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina

[2]Institute of Biomedicine, Faculty of Medicine, University of Turku, Turku, Finland

[3]Department of Computer Science, College of Engineering, Effat University, Jeddah, Saudi Arabia

Correspondence
Adaleta Gicić, Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina.
Email: adaleta.gicic@etf.unsa.ba

Summary

Credit scoring is one the most important parts of credit risk management in reducing the risk of client defaults and bankruptcies. Deep learning has received much attention in recent years, but it has not been implemented so intensively in credit scoring compared to other financial domains. In this article, stacked unidirectional and bidirectional LSTM (long short-term memory) networks as a complex area of deep learning are applied in solving credit scoring problems for the first time. The proposed robust model exploits the full potential of the three-layer stacked LSTM and BDLSTM (bidirectional LSTM) architecture with the treatment and modeling of public datasets in a novel way since credit scoring is not a time sequence problem. Attributes of each loan instance were transformed into a sequence of the matrix with a fixed sliding window approach with a one-time step. Our proposed models outperform existing and much more complex deep learning solutions thus we succeeded in preserving simplicity. In this article, measures of different types are employed to carry out consistent conclusions. The results by applying three hidden layers on the German Credit dataset showed an accuracy of 87.19%, for Kaggle dataset accuracy reached 93.69%, and for Microcredit dataset accuracy of 97.80%.

KEYWORDS

credit risk management, credit scoring, deep learning, stacked BDLSTM, stacked LSTM

1 | INTRODUCTION

Credit scoring is one of the most important parts of an effective credit risk assessment. Credit scoring is not a new concept and was first proposed decades ago by Durand[1] in order to determine if the applicant is credit-worthy or not. It is regarded as a classification problem[2] and the dataset used in that process usually consists of the applicant's demographic and transactional data and the characteristics of the credit that it applies for. Credit scoring models have evolved significantly over the years, together with the development of new algorithms, increased processor power, the appearance of powerful database platforms, big data and so forth. Various credit scoring models have been proposed in the past, from discriminant analysis to probit or logistic regression models, decision trees, k-nearest neighbors, neural networks, SVM (support vector machine), the ensemble of algorithms, to deep learning techniques. Despite the fact that credit scoring was one of the first fields of economics in which machine learning methods were implemented, most banks are still bonded to linear regression models due to the rigorous European[3] and global banking regulatory standards that oblige them to explicitly explain credit score decisions to customers and regulators.[4] Thus, "black box" models that are not explainable are one of the major issues that still limit the utilization of advanced credit scoring models, especially those implemented by ensemble methods.

In many credit risk evaluation domains, deep learning (DL) is superior to traditional machine learning methods, and classifier ensembles perform significantly better than single classifiers.[5] However, the credit evaluation models based on a deep learning ensemble algorithm have not been studied enough.[5] Comprehensive surveys of DL techniques applied to credit scoring show that some DL techniques have never been employed in the improvement of credit scoring models until now.

The LSTM (long short-term memory) based model on the classical German credit dataset was first applied in 2021,[5] but the stacked LSTM technique and BDLSTM (bidirectional LSTM) however, have not been applied for credit scoring prediction by now.[6]

This article comprises the following contributions.

- The main contribution of this study is to utilize the multilayer unidirectional and bidirectional long short-term memory architecture for credit scoring. To the best of the authors' knowledge, there has been no study on employing those two techniques in credit scoring. Credit scoring models that utilize Stacked LSTM, and particularly Stacked BDLSTM outperform other analyzed techniques. The influence of a different number of hidden layers applied in stacked LSTM and BDLSTM was also analyzed and compared in this study.
- This research will also address the problem of proper dataset modeling. In previous studies, details on data shaping in most of the cases were not provided. Not only is it adequate data shape that yields superior results, but this approach also proposes a less complex solution, without the need for additional boosting or data oversampling. Details on data preparation are described in Section 4.3.
- The third contribution is to diminish shortcomings recognized in some of the previous studies regarding using insufficient and limiting datasets that can pose a generalizability threat. However, some works such as[7,8] applied various and rich credit datasets for verification of the performance of proposed model such as: Australian, German, Japanese, PaiPaiDai, and GMSC datasets. In this article datasets of different sizes and origins, with different numbers and types of attributes are used: German, Kaggle, and Microcredit datasets. The absurdity was that most of the published works used big data machine learning methods on extremely small datasets.[9] It is not good practice to conclude that one method is better than another based only on one dataset over one period.[9]
- Another contribution is to overcome the limitations of public datasets that the studies rely on. Datasets are usually not as new as desired and were mostly created before the financial crisis. It would be useful to model the behavior of the institutions under the new regulations as well as the economic recovery seen later.[10] That gap is also closed within this research because unlike the German dataset, the Kaggle dataset and the Microcredit dataset are much newer.
- The goal of this article is also to demonstrate and validate performance improvements based on real circumstances. Furthermore, all the techniques were applied to an unseen dataset with extraordinary results for all three datasets.

The rest of the article is structured as follows. Section 2 gives insight into the previous relevant research in the field and reviews the related work. Section 3 describes the data structure, theoretical foundations and evaluation metrics used in this paper. In Section 4 proposed enhancements regarding deep learning model architectures, techniques, novel dataset modeling, and methods used in the research are explained. Section 5 discusses the results of the experiments and comprehensive comparison to recent methodologies. Section 6 presents the concluding remarks of the study.

2 | LITERATURE REVIEW

Methods applied in credit scoring can generally be divided into statistical and machine learning methods.[2] A large number of papers have been published in the financial sector, and much research on the application of machine learning has been ongoing in recent years. Such a trend is evident in various disciplines in credit risk management and the improvement of credit scoring applications as well.[11] The most comprehensive analysis of the application of statistical and machine learning models in credit scoring was performed by Lessman,[12] with the conclusion that there is no consensus on the best performing algorithm. However, the majority of the research proved that machine learning techniques are significantly better than any statistical model.[5] Recently, a lot of papers on the application of deep learning in all financial disciplines have been published, but not enough in solving credit scoring problems.

2.1 | LSTM techniques in credit scoring

Publications and research regarding the application of deep learning (DL) in credit scoring applied to classical static datasets are still in their early stages. Therefore, an important research problem is to further improve the performance of credit scoring based on deep learning technology.[5,13] One of the newer surveys on deep learning applications[9] does not mention any of the concrete state-of-the-art deep learning algorithms applied to credit scoring datasets. According to the review,[6] none of the studies have covered the LSTM deep learning technique on credit scoring models, but it is evident that a considerable number of papers that cover LSTM other financial fields such as algo-trading, fraud detection, financial distress, bankruptcy, bank risk, mortgage risk, crisis forecasting studies, portfolio management, asset pricing, and derivatives market studies, crypto currency, and block chain studies, and financial sentiment studies coupled with text mining for forecasting. The reason could be found in the fact that most of the datasets used in fields other than credit scoring are transactional, time sequence data, what is ideal for LSTM utilization. Another paper[14] uses LSTM in credit scoring but not stacked LSTM and not on classical public German or Australian dataset but on a massive transactional dataset with 200 million transactions and over 740.000 clients.

A recent study[5] has applied an ensemble model with LSTM as a base technique on classical credit scoring datasets, that is, on German and Taiwan datasets. Additionally, it used SMOTE (synthetic minority oversampling technique) and AdaBoost and was generally more competitive when addressing imbalanced credit risk evaluation problems than other models.[5] Zhang et al.[15] also applied a single LSTM network but on peer-to-peer lending credit scoring evaluations, which resulted in better performance compared to statistical and machine learning models. Another study on credit scoring with standard LSTM application was published with improved prediction accuracy, but it was applied to an anonymous P2P lending platform in China rather than the standard credit scoring dataset.[16] Mahajan et al.[17] applied deep feed forward neural networks (DFNNs) that uses family of nonlinear neural network activation functions. To the best of our knowledge, Stacked LSTM and BDLSTM as novel techniques have never been applied in credit scoring before.

2.2 | Other deep learning techniques in credit scoring

A comprehensive survey with a focus on deep learning employment in financial applications[6] concludes that the number of research projects increases every year in an accelerated fashion since the financial industry and academia recognize the promises and possibilities that deep learning offers in various application fields. A large number of various deep learning technologies have been applied in credit scoring recently[6]: deep belief networks (DBN),[18] auto encoders (AE),[19] restricted boltzmann machines (RBM),[20,21] deep neural network (DNN), deep convolutional neural networks (DCNN),[22] boosted DNN, particle swarm optimization (PSO)-back-propagation artificial neural networks (BP-NN)[23] convolutional neural network (CNN), multilayer perceptron (MLP), and deep MLP (DMLP). The DCNN significantly outperformed DMLP.[22] DMLP fits well for both regression and classification problems, and RNN (recurrent neural network)-based models can work directly with time-varying data, making it easier for researchers to develop DL models.[24] Deep neural network (DNN) with RF, bagging, and boosting methods were compared[15] and the results showed that the ensemble methods performed better than DNN. CNN's (convolutional neural network) application in the financial domain has become a recent trend with the innovative transformation of 1-D time-varying financial data into 2-D stationary image-like data. It was demonstrated that CNN deep learning models perform better compared to statistical and classical machine learning models.[25] Successful credit scoring models were proposed utilizing the power of the CNN method and relief algorithm by converting data sets into images by bucketing features and mapping them into image pixels. The source of the dataset was a Chinese consumer finance company, not the classical German or Australian Credit dataset.[26] The proposed model was however applied only to numerical features and this gap is closed by a new study[27] that used both categorical and continuous features. Another study on CNN application in credit default prediction using the Lending Club dataset demonstrated superior performance regarding the accuracy and AUC (area under the curve).[28] Pławiak et al.[29] proposed a 29-layer network with different machine learning algorithms (deep genetic hierarchical network of learners) with limitations in extreme complexity and the long-term training. Ensembles of algorithms were also efficiently combined and applied to the German Credit Score dataset, such as the incremental learning ensemble method (ILEM)[30] and CF-GA-Ens (clustering with fuzzy assignment—genetic algorithm—ensemble learning)[31] and the novel tree-based overfitting-cautious heterogeneous ensemble model (OCHE).[32]

Some of the newer studies also applied novel machine learning techniques to German credit scoring models: step-wise multi-grained augmented gradient boosting decision trees,[33] dynamic 1-nearest neighbor[34] and Cost-sensitive Neural Network Ensemble.[35]

3 | DATASETS AND METHODOLOGY

In this section, datasets used, and a brief introduction of algorithms and theoretical foundations applied in this study are described. In dealing with complex financial data, two novel deep learning models have been experimented (stacked LSTM and stacked bidirectional LSTM) and applied to three credit scoring datasets: the classical German credit score dataset with a small number of instances, the large-sized Kaggle dataset, and the middle-sized Microcredit dataset collected from the live system.

Deep learning is a technique based on artificial neural networks with many layers that enables computers to imitate human behavior. LSTM and BDLSTM are RNN-based models that are designed to predict the next value of the time series, whereas our problem is treated as a classification, and therefore the most important factor was the data set fitting into neural networks. The following sections highlight the deep learning models and datasets that are used in this research.

3.1 | Datasets

In this research, three types of datasets are used: German Credit, Kaggle, and Microcredit. German and Kaggle datasets are publicly available datasets from the banking sector and we used them to transparently compare our results to previous ones. German credit dataset is obtained from the UCI machine learning repository, and a detailed explanation of the dataset can be found on.[36] A publicly available Kaggle dataset can be found on.[37] The Microcredit dataset is collected from the live system and used in the study.[38]

A general preview of the data sets used in this paper is given in Table 1.

Figure 1 shows that datasets have different imbalance ratios.

Various surveys of state-of-the-art classification algorithms for credit scoring[12] emphasized the need to overcome limitations by testing multiple methods on multiple datasets, particularly because the various datasets have various data structures.[12]

There are multiple reasons for introducing the three different data sets. German dataset is chosen since it is used in most of the research in the credit scoring field. Therefore, evaluations and improvements of the proposed model are more relevant and could be empirically confirmed. Second, to overcome limitations recognized in other studies by using scanty and restrained datasets that could not assure generalizability and to demonstrate and validate improvement based on the real circumstances. Third, to overcome the deficit of newer credit instances in public datasets usually collected before the financial crisis.[10] Deep learning models require more data for training to avoid overfitting[39] which is assured by introducing the Kaggle and Microcredit datasets. Fourth, mostly applied credit scoring data sets are not entirely appropriate for deep learning state-of-the-art algorithms, which reach their full potential only by using larger data sets.[9]

3.1.1 | Descriptive statistics of the datasets

Descriptive statistics organize and describe characteristics of attributes in the data sample, measure the most common patterns and analyze the dispersion of the distribution of the dataset. So, descriptive statistics is divided into two parts:

- Measures of central tendency (mean, median, and mode)
- Measures of variability or dispersion (standard deviation, variance, minimum and maximum variables, quartiles, kurtosis, and skewness)

TABLE 1 Preview of credit scoring datasets.

Dataset	Number of instances	Good credit	Bad credit	Number of attributes	Imbalance ratio
German data	1.000	700	300	20	2.33
Kaggle data	150.000	139.974	10.026	10	13.96
Microcredit dataset	14.387	8.642	5.745	25	1.50

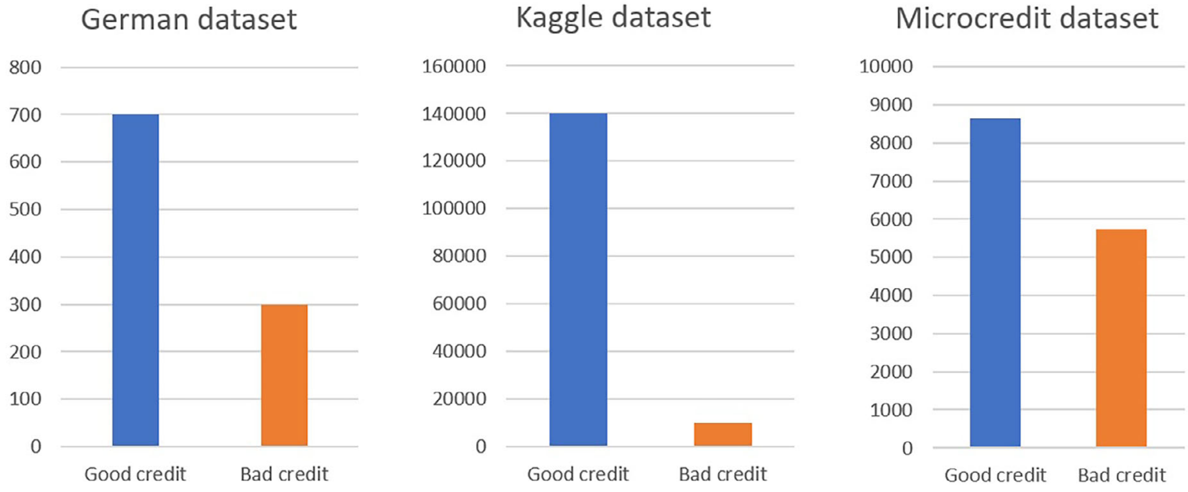

FIGURE 1 Credit dataset structure: German, Kaggle and Microcredit. (1) Good credit, (2) Bad credit.

Measures of central tendency is a value that describes the center point of the dataset, while measures of dispersion describe the shape and distribution of data within a dataset.

Descriptive statistics for datasets used in this paper are given in Table 2 for German dataset, Table 3 for Kaggle data and Table 4 for Microcredit dataset.

3.2 | Stacked long short term memory

LSTM networks are a special type of RNN structured to remember and predict based on short and long-term dependencies that are trained with time series data and to deal with the vanishing gradient problem.[40] LSTM weight updates and preferred optimization methods work on the same principle as RNN. Use of hyperparameters: number of hidden layers, number of units in each layer, network weight initialization, activation functions, learning rate, momentum values, number of epochs, batch size, decay rate, optimization algorithms, sequence length for LSTM, gradient clipping, gradient normalization, and dropout are also the same.[41]

LSTM networks consist of LSTM units composed of three special multiple cell gates: input, output, and forget gate.[40] The purpose of gates is to regulate the flow of information, with each cell remembering the desired values over arbitrary time intervals. Each neuron assigns a value, and all connections assign weights. Neuron values are calculated by mathematical functions, taking into account the weights and values of previous layers.[25] Figure 2 shows one LSTM cell.

Units are then merged to form an LSTM layer.[24] Although primarily developed to deal with sequential data, the LSTM network is very effective in mining the interrelationships between credit data variables.[5] Stacked LSTM novel architecture is described in Reference 43. In a stacked multi-layer LSTM architecture, the output of a hidden layer will be fed as the input into the subsequent hidden layer.[44] Stacked LSTM differs from the original LSTM in the number of hidden layers where each layer contains multiple memory cells. Formulas 2–7 explain the form of the forward pass of the LSTM unit[44]:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right), \tag{1}$$

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right), \tag{2}$$

$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right), \tag{3}$$

$$\tilde{c_t} = tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right), \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}, \tag{5}$$

$$h_t = o_t * tanh\left(c_t\right), \tag{6}$$

where i_t stands for input gate, f_t represents forget gate, o_t output gate, x_t represents current timestamp, h_{t-1} represents LSTM output of the previous timestamp, w represents weights, σ represents sigmoid function, tan$h$ represents hyperbolic tangent function and * is the element-wise vector/matrix multiplication operator.

3.3 | Stacked BDLSTM

Stacked BDLSTM evolved from the principle of Bidirectional RNNs (BRNNs) introduced by[45] to handle input sequences whose beginnings and ends are known in advance. In addition to BRNNs, BDLSTM is also a successor of bidirectional LSTMs which takes both context information by concatenating left and right summary vectors.[46,47] BDLSTM as an extension of the traditional LSTM architecture consists of numerous memory cells. The concept of bidirectional LSTM is based on reading the training data in two-time directions by training the neural network.[47] In sequential classifications, only the left context is considered, so a summary vector is created by reading a series of inputs from left to right. Because bidirectional LSTM takes both the left and right context information, it can yield higher predictive performance compared to unidirectional deep neural architectures.[46] Architectures with several hidden layers are generally more efficient and can progressively build up a higher level of representation of sequence data.[43] The number of layers can vary, and the architecture can also consist of a mix of BDLSTM and LSTM layers. The first feature learning layer, for example, can be BDLSTM and the other can be LSTM, but since BDLSTM contains more learnable parameters, the architecture of stacked BDLSTMs has the potential to perform better.[43]

TABLE 2 Descriptive statistics for German dataset.

Attributes	Mean	STD	Min	Q1 (25%)	Q2 (50%)	Q3 (75%)	Max	Median	Mode	Variance	Kurtosis	Skew
Risk	0.7	0.45848687	0	0	1	1	1	1	1	0.21021	−1.23828	−0.87418
Account balance	2.577	1.257637727	1	1	2	4	4	2	4	1.581653	−1.6637	0.006957
Duration of credit (month)	20.903	12.05881445	4	12	18	24	72	18	24	145.415	0.919781	1.094184
Payment status of previous credit	2.545	1.083119637	0	2	2	4	4	2	2	1.173148	−0.57906	−0.01189
Purpose	2.828	2.74443946	0	1	2	3	10	2	3	7.531948	0.554083	1.178887
Credit amount	3271.248	2822.75176	250	1365.5	2319.5	3972.25	18424	2319.5	1258 1262 1275 1393 1478	7,967,927	4.292481	1.949594
Value savings/stocks	2.105	1.580022617	1	1	1	3	5	1	1	2.496471	−0.68022	1.016677
Length of current employment	3.384	1.208306254	1	3	3	5	5	3	3	1.460004	−0.93433	-0.11761
Installment percent	2.973	1.118714674	1	2	3	4	4	3	4	1.251523	−1.21047	-0.53135
Sex & marital status	2.682	0.708080064	1	2	3	3	4	3	3	0.501377	−0.00257	−0.30515
Guarantors	1.145	0.477706189	1	1	1	1	3	1	1	0.228203	9.328756	3.264249
Duration in current address	2.845	1.103717896	1	2	3	4	4	3	4	1.218193	−1.38145	−0.27257
Most valuable available asset	2.358	1.050208998	1	1	2	3	4	2	3	1.102939	−1.23852	0.045673
Age	35.542	11.35267013	19	27	33	42	75	33	27	128.8831	0.620529	1.024712
Concurrent credits	2.675	0.705601072	1	3	3	3	3	3	3	0.497873	1.512588	−1.82652
Type of apartment	1.928	0.530185908	1	2	2	2	3	2	2	0.281097	0.484031	−0.07383
No of credits at this bank	1.407	0.577654468	1	1	1	2	4	1	1	0.333685	1.604439	1.272576
Occupation	2.904	0.653613962	1	3	3	3	4	3	3	0.427211	0.501891	−0.37429
No of dependents	1.155	0.362085772	1	1	1	1	2	1	1	0.131106	1.649274	1.909445
Telephone	1.404	0.490942996	1	1	1	2	2	1	1	0.241025	−1.85014	0.391868
Foreign worker	1.037	0.188856206	1	1	1	1	2	1	1	0.035667	22.1822	4.913027

TABLE 3 Descriptive statistics for Kaggle dataset.

Attributes	Mean	STD	Min	Q1 (25%)	Q2 (50%)	Q3 (75%)	Max	Median	Mode	Variance	Kurtosis	Skew
seq	75000.5	43301.41	1	37500.8	75000.5	112500.3	150,000	75000.5	-	1.88 E+09	−1.2	0
SeriousDlqin2yrs	0.06684	0.249746	0	0	0	0	1	0	0	0.062373	10.0331	3.468857
Revolving Utilization Of Unsecured Lines	6.048438	249.7554	0	0.02987	0.154181	0.559046	50,708	0.15418	0	62377.75	14544.71	97.63157
Age	52.29521	14.77187	0	41	52	63	109	52	49	218.208	−0.49467	0.188995
Number Of Time 30-59 DaysPastDueNotWorse	0.421033	4.192781	0	0	0	0	98	0	0	17.57941	522.3765	22.59711
DebtRatio	353.0051	2037.819	0	0.17507	0.366508	0.868254	329,664	0.36651	0	4,152,704	13734.29	95.15779
Monthly Income	5347.941	13152.14	−1	1550	4357.5	7400	3 E+06	4357.5	−1	1.73 E+08	22426.11	119.9039
Number Of Open Credit Lines And Loans	8.45276	5.145951	0	5	8	11	58	8	6	26.48081	3.091067	1.215314
Number Of Times 90 Days Late	0.265973	4.169304	0	0	0	0	98	0	0	17.38309	537.7389	23.08735
Number Real Estate Loans Or Lines	1.01824	1.129771	0	0	1	2	54	1	0	1.276382	60.47681	3.482484
Number Of Time 60-89 Days Past Due NotWorse	0.240387	4.155179	0	0	0	0	98	0	0	17.26552	545.6827	23.33174
Number Of Dependents	0.711253	1.135585	−1	0	0	1	20	0	0	1.289554	2.815182	1.489012

TABLE 4 Descriptive statistics for Microcredit dataset.

Attributes	Mean	STD	Min	Q1 (25%)	Q2 (50%)	Q3 (75%)	Max	Median	Made	Variance	Kurtosis	Skew
Monthly available amount	0.04587	1.74109	−1	−1	−1	2	3	−1	−1	3.03140016	−0.81839	1.07773
Duration of credit (month)	14.1751	7.57574	0	10	12	18	60	12	12	57.3919054	5.00929	1.7176
Credit history	0.77619	1.28361	0	0	0	1	4	0	0	1.6476659	1.58868	1.69434
Purpose	1.78237	0.56249	1	1	2	2	5	2	2	0.31639162	2.35284	0.38008
Credit amount	2111.4	2125.49	0	1000	1500	3000	#####	1500	1000	4517701.55	17.7329	3.08418
Gender	1.01786	0.17257	0	1	1	1	2	1	1	0.02977959	29.5781	3.16507
Employment status	1.96365	0.18717	1	2	2	2	2	2	2	0.03503321	22.5546	−4.95494
Credit amount.1	86.0954	264.823	−1	−1	−1	0	8718	−1	−1	70131.342	127.661	7.56244
Type of customer	0.99652	0.05885	0	1	1	1	1	1	1	0.00346352	282.842	16.8761
Guarantors	92.8798	502.857	−1	1	14	146	####	14	14	252864.964	540.413	21.9395
Reprogrammed	0	0	0	0	0	0	0	0	0	0	0	0
Family size	0.82644	2.13319	−1	1	0	3	14	0	−1	4.55050897	−0.57998	0.78532
Age	47.3163	12.4196	18	38	48	56	115	48	50	154.246978	0.11295	0.18128
Marital status	1.10162	0.95896	0	1	1	1	4	1	1	0.91960463	2.16124	1.4578
Housing	0.86189	0.98651	0	0	0	2	3	0	0	0.97320825	−1.86371	0.29847
Number of active credits in this company	0.74824	0.75333	0	0	1	1	4	1	1	0.56750613	0.48813	0.83839
Number of all credits in this company	5.03774	5.68243	0	1	3	6	47	3	1	32.2900393	8.60375	2.53335
Nationality	2.82915	1.83525	1	1	2	5	5	2	5	3.36813944	−1.80681	0.25781
Telephone	1.98109	0.1362	1	2	2	2	2	2	2	0.01854981	47.9297	−7.06563
Credit type	1.13755	0.44609	1	1	1	1	3	1	1	0.19899758	10.0197	3.31133
Business type group	2.22798	1.25267	0	1	2	3	4	2	1	1.56917933	−1.39282	0.09655
Number of school kids	−0.71738	0.7466	−1	−1	−1	−1	6	−1	−1	0.55740875	8.74735	2.94726
Business duration	0.15549	2.18931	−1	−1	−1	−1	5	−1	−1	4.79307481	0.26883	1.45259
Number workers	−0.68423	2.05087	−1	−1	−1	−1	200	−1	−1	4.20606503	7207.64	79.4559
Credit status	0.4233	0.4941	0	0	0	1	1	0	0	0.2441339	−1.90385	0.31051
Class	0.39932	0.48978	0	0	0	1	1	0	0	0.23987998	−1.83118	0.41119

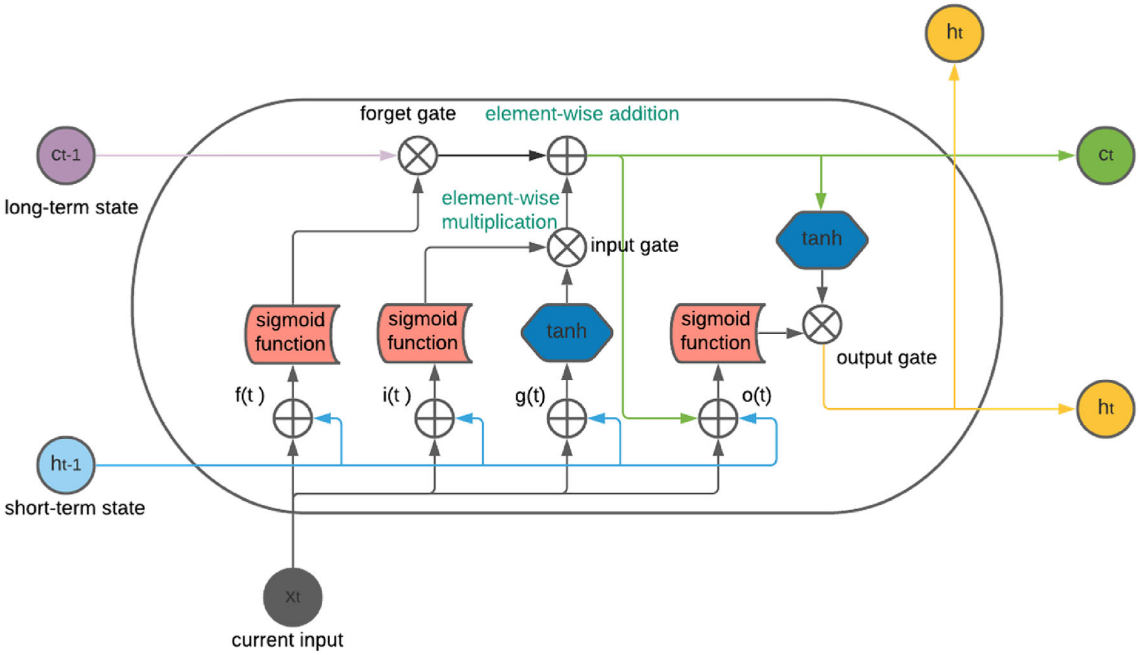

FIGURE 2 Architecture of LSTM.[42]

3.4 | Evaluation metrics

To assure effective model evaluation and a better understanding of model performance, various indicators are chosen in this paper. Chosen metrics measure different trade-offs in predictive performance.[48] In this paper, we measured the classifier performance by calculating metrics from different categories, some of which are widely used in the literature: Accuracy, Precision, Recall, and F1.

$$Accuracy = \frac{TP + TN}{P + N}, \tag{7}$$

$$Precision = \frac{TP}{TP + FP}, \tag{8}$$

$$Recall = \frac{TP}{TP + FN}, \tag{9}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{10}$$

TP, number of true positives.
FP, number of false positives.
TN, number of true negatives.
FN, number of false negatives.
P, number of positives.
N, number of negatives.

The F1 score was used for better insight into the performance of the model since it comprises both precision and recall. Accuracy, on the other hand, is the most popular metric that calculates the ratio of correctly classified examples out of all examples, and it is easily understandable for overall performance irrespective of the concrete class. Results for Precision, Recall, and F1 within this paper cover both, macro and weighted averages, and Accuracy is a micro averaging since it is the ratio of correct predictions and the whole sample. Our dataset is imbalanced thus, a weighted average indicator is more relevant than a macro average since it assigns greater contribution to classes with more observations in the dataset.

Accuracy is used instead of AUC since datasets are not highly imbalanced, with an imbalanced ratio of 2.33 for German data, 13.96 for Kaggle data, and 1.50 for the Microcredit dataset. Business costs for FP (false positive) and FN (false negative) predictions are approximately the same since FP leads to credit default and brings a loss to the financial institution and FN results in the rejection of the client that could pay off the credit and bring profit to the financial institution. Credit scoring datasets are not significantly imbalanced datasets compared to fraud detection datasets in which case the AUC measure is used for the model's evaluation. The aim of such predictive models is for minority class that is, fraudulent transactions to be correctly distinguished and their occurrence is even less than 1%. However, accuracy and AUC are both helpful in model estimation and in comparing one model to another.

4 | PROPOSED ENHANCEMENTS

The following section highlights the deep learning models used in this research, data processing and fitting, and the technology stack. Two types of enhancements are presented. The first enhancement refers to the setup of hyperparameters for both networks. Choosing the most appropriate architecture for credit score predictions is a challenging task, but before applying the dataset to the neural network, the original data has to be adjusted and put into context. Thus, the second and more important enhancement is related to dataset manipulation, since these kinds of networks are designed to predict the next value of a time series, whereas our problem is not time dependent.

4.1 | Proposed stacked LSTM and BDLSTM credit scoring framework

Accurate models have to be adapted to the structure of the domain since the real data typically differ in practice and it is almost impossible to establish and define universal learners.[49] Our model is a four-layer stacked LSTM and BDLSTM with 60 units in the input layer, 60 units in the first hidden layer, 80 units in the second 80, and 120 units in the third layer. The third hidden layer is further connected to the dense Layer and mapped to the output. An LSTM layer creates a sequence output rather than a single value output to the LSTM layer below. Since LSTM operates on sequence data, the addition of layers adds levels of abstraction of input observations over time. The depth of the network, however, was more important and showed better performance than the increase in the number of memory cells while modeling a given layer[50] .The model that we used is described in Figure 3.

The stacked Bidirectional LSTM (BDLSTM) network is shown in Figure 4.

BDLSTMs can make use of both forward and backward dependencies. The features are passed through pooling and flattening the layers before being passed to the next layer. For the German dataset, this method of independently extracting features from 20 different units of 1 timestamp outperformed the conventional method of passing the chunks of timestamps. The same was performed with the Kaggle and Microcredit datasets, with a different number of units.

The complexity of the proposed model increases together with the number of input parameters. Therefore, the total number of parameters that the model contains goes from 190,201 parameters for German Credit LSTM based models to 526,321 parameters for the Microcredit BDLSTM model. For the test data set, the threshold is a very important subject to be set correctly. Empirical results show that the cut-off is ideally set to 0.64 for both LSTM and BDLSTM and for all three datasets, with an exception for the Kaggle dataset and BDLSTM based model, with a threshold of 0.57 which minimized error in misclassification. All models were evaluated with the same shuffling, same train, test, and validation set to maintain consistency. The stacked bidirectional LSTM (BDLSTM) algorithm outperformed the stacked LSTM since the BDLSTM allows a two-way flow of information.

4.2 | Hyperparameter optimization

Even the smallest adaptation of hyperparameters values can sometimes lead to significant degradations in LSTM or BDLSTM model performance. To avoid a biased approach in defining hyperparameters, model selection must be performed again each time the model is fitted to a new dataset.[5] Some of the most important hyperparameters are the number of nodes and hidden layers, number of units in a dense layer, dropout, weight initialization, decay rate, activation function, learning rate, momentum, number of epochs, and batch size. The smallest change in hyperparameters, can sometimes lead to significant performance degradations.

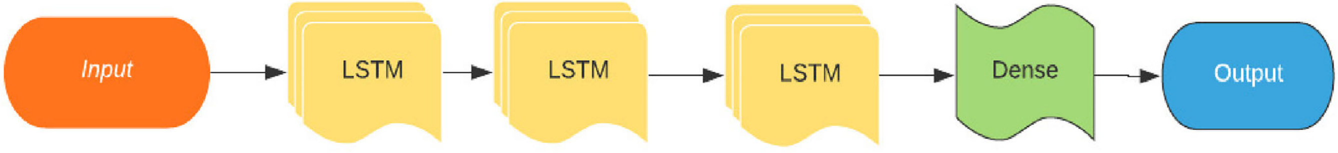

FIGURE 3 Architecture of stacked LSTM.

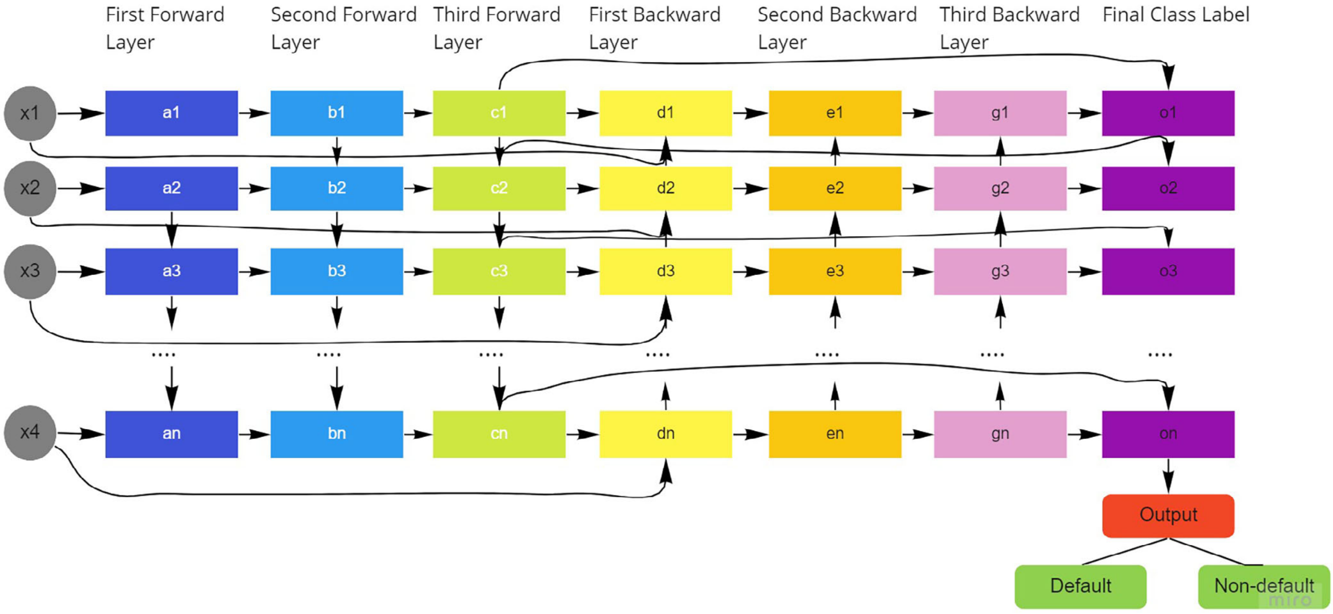

FIGURE 4 The three-layer stacked bidirectional LSTM.

1. The LSTM and BDLSTM architectures proposed in this article are designed as three-layer models with 60 nodes for the input layer. The first layer consists of 60 nodes, the second of 80, and the third of 120 nodes. This means that our model consists of three hidden layers, that is, layers of nodes between the input and output layers. The third layer is connected further with the Dense layer.
2. A Dense layer is a most frequently used layer where each neuron receives input from all neurons in the previous layer. The number of units in the Dense layer is one since as the output we have information nondefault/default (1/0).
3. A Dropout parameter in our experiment is set to 0.2, which is a widely accepted value. Each layer of LSTM and BDLSTM layer except the output layer is paired with the Dropout layer which minimizes overfitting in training and improves generalization. It is not a good practice though to add Dropout to the output layer.
4. Weights are usually initialized randomly to small numbers which indicate the importance of an input value. With a technique called Gradient Descent, neurons are adjusted with each iteration which reduces the cost function[25]
5. The Decay rate is by default set to 0.97 which is also done in our setup. This factor makes weights decay to zero.
6. The selection of the activation function depends on the type of problem. The Relu function reduces the vanishing gradient problem, but not completely. Regarding the activation function between hidden layers, we set Relu function for all layers. The LSTM cell itself contains three sigmoid functions and one tangent function, except for the Dense layer.
7. Learning rate is a hyperparameter that defines how fast networks converge or even diverge. A lower rate slows down learning and steps towards the minimum loss function. The default learning rate of 0.01 is set.
8. Momentum accelerates gradient descent that accumulates a velocity vector in the direction of persistent reduction in the objective across iterations.[51] The momentum value in this experimental setup was left as default that is, no momentum was set.
9. The number of epochs we used in our experiment is 50. Epochs represent the number of iterations for the dataset. Lots of epochs are mandatory for algorithm to learn all those subtle features, but too many epochs can lead to overfitting of the training dataset, whereas too few may result in an underfit model.
10. The batch size defines the number of input–output pairs that are passed before updating internal model parameters. It must not be confused with window size. The batch size in our case is set at 32.

4.3 | Dataset preprocessing

The effectiveness of machine and deep learning models depends largely on the nature and quality of datasets, but mostly on the way the dataset is analyzed, adapted, processed, and applied.[52] In this research, data processing encompassed the following steps: missing values handling, encoding, scaling, shuffling, dataset splitting, outlier detection, and data modeling. In the specific cleaning process, uniqueness, completeness, validity, relevance, timeliness, and consistency had to be ensured.[53]

Scaling is performed for each attribute for all three training datasets by using the MinMaxScaler algorithm. Values are scaled between zero and one. MinMaxScaler performs better than the standard scalar, particularly in cases where the distribution is not Gaussian, or the standard deviation is very low but could pose problems if outliners in the dataset are detected or if unseen samples of data fall out of the range of training data variables.[54] It was not the case with our datasets.

With the aim of authentic and accurate evaluation of the proposed model, datasets are split into three data sets: training, testing, and out-of-sample datasets. Foremost, the out-of-sample dataset was completely separated so that we could afterward assess the efficiency of the proposed model on that "unseen dataset". Therefore, the separated dataset would not have any of the roles in the process of model fitting.

Classes within our datasets are not evenly sorted. To assure even distribution, we had to perform shuffling of datasets before data separation for out-of-sample instances. At that point, the dataset was ready for splitting.

To obtain the average value, 5-fold cross-validation was used. It provides relatively stable results, and it is computationally cheaper.

Correlation between variables is also checked using various tools. Feature selection and SMOTE[38] techniques used in most of the previous studies were not performed in this research since we achieved high accuracy on both trained and unseen datasets, despite the fact that datasets, particularly German Credit, are extremely small in size.

4.4 | Novelty in dataset modeling

Credit scoring data sets are not time series, and one row consists of aggregated attributes that usually contain historical data related to the previous behavior, loan information, personal, financial, employment information, and default (1) and nondefault (0) labels.

Furthermore, dataset shapes are not formed on the same principle as in the previous studies. The input for LSTM and BDLSTM based models is based on sliding window chunks that consider past events, but the nature of credit scoring data sets is different. The features of credit were treated as time series data, and the next value to predict is the classification into default and nondefault. The sequences of credits are not dependent on each other in time perspective.

In this study, data shaping is performed by segmentation of raw data with a fixed sliding window approach. Each data point was a three-dimensional array with one time-step and 20, 10, and 25 features for each dataset respectively, German, Kaggle, and Microcredit. Overlapping of time series data is avoided this way since the nature of this dataset is not time dependent. While shaping in standard time sequence way with sliding windows and going back in history, the experiment shows that the performance of our model deteriorates since the nature of our dataset is not time sequence of events and does not resemble stock prediction or in fraud detection dataset for example. Thus, credit loan attributes were sequenced instead of the traditional approach where credit loans themselves are sequenced, since in practice, there is no such time-dependent correlation between individual credit loans.

Training data shape for the German Credit dataset is (937, 1, 20) and 21st column represented result set, and each of the 21 units is passed independently into the LSTM layer. A number of 63 instances is separated prior to training to form an independent out-of-sample dataset for testing purposes. For Kaggle training dataset shape is (130,000, 1, 10) and 11th column. Each of the 11 units is passed independently. The set of 20.000 instances is divided into unseen test datasets. On the same principle, for the Microcredit dataset, the train shape is (13,000, 1, 25) and 26th column is the result and each of 26 units is passed independently. A total of 1387 instances are extracted into the unseen dataset.

Even though German Credit is a dataset of extremely small size to be applied on these kinds of algorithms and due to superior results and generalization with high accuracy on out-of-sample datasets, additional techniques such as SMOTE and AdaBoost applied in the study[5] are omitted on purpose to demonstrate the efficiency of proposed Stacked Unidirectional and Bidirectional LSTM models.

4.5 | Technology stack

Deep learning models in this study are implemented in Python using the high-level neural networks API Keras (application programming interface), which is running on top of TensorFlow. Even though open-source libraries such as Keras allow us to write complex code and produce robust deep learning models, it is still our responsibility to choose the right methods, set hyperparameters for a particular problem, where even the smallest improvement in loss can result in superior performance.

5 | RESULTS AND DISCUSSION

In this section, results achieved by applying deep learning algorithms will be presented and analyzed and their effectiveness will be assessed by a set of different measures. Experiment results in this section demonstrate the superior performance of Stacked LSTM and BDLSTM techniques.

TABLE 5 German credit scoring model performance.

	Stacked LSTM	Stacked BDLSTM
Accuracy (training)	82.28%	87.19%
Accuracy (testing)	84.13%	88.89%
Precision (macro avg)	79%	85%
Precision (weighted avg)	84%	87%
Recall (macro avg)	81%	83%
Recall (weighted avg)	83%	87%
F1 (macro avg)	80%	84%
F1 (weighted avg)	83 %	87%

TABLE 6 Kaggle credit scoring model performance.

	Stacked LSTM	Stacked BDLSTM
Accuracy (training)	93.72%	93.71%
Accuracy (testing)	93.53%	93.34%
Precision (macro avg)	81%	78%
Precision (weighted avg)	92%	92%
Recall (macro avg)	54%	56%
Recall (weighted avg)	93%	93%
F1 (macro avg)	56%	59%
F1 (weighted avg)	91%	91%

Data shaping, together with proper hyperparameter configuration, plays a crucial role in the leverage of a model's efficiency. Our methods performed well with an imbalanced dataset, with the highest accuracy of 87.19%, 93.71%, and 97.79% for German Credit, Kaggle Credit, and Microcredit, respectively.

For consistency, our stacked LSTM and BDLSTM based models are designed with three hidden layers for all three datasets. The influence of a different number of hidden layers is also analyzed and compared. Experimental results regarding the influence of performance depending on the number of hidden layers revealed that there is no universal solution for all models and all datasets.

Proposed novel architectures are empirically compared to other methods to show significant success in applying them to publicly available German Credit and Kaggle datasets and to Microcredit datasets.

The models we proposed based on stacked neural networks are trained by minimizing the mean square error (MSE) using the Adam optimization method and RMSE metrics. The features are passed through pooling and flattening layers before being passed to the next layer.

5.1 | Results applying stacked LSTM and BDLSTM to three different datasets

The performance of the German credit scoring model applying novel methodology using different metrics is shown in Table 5.

Kaggle credit scoring model performance applying novel methodology using different metrics is shown in Table 6.

Table 7. shows the performance of a microcredit scoring model using novel methodology and different metrics. Classes from the original Microcredit dataset: good-A, bad-E, and poor-C classes were transformed into two microcredit classes where E and C classes were merged into one in order to create two classes model, since German Credit and Kaggle are also two class models.

5.2 | Influence of the number of hidden layers

Within this study, the performance of the proposed model by increasing and decreasing the number of hidden layers of chosen networks was also experimented with. A three-layer structure for both Stacked LSTM and BDLSTM based models was accepted as the baseline since it gives the best

TABLE 7 Microcredit scoring model performance.

	Stacked LSTM	Stacked BDLSTM
Accuracy (training)	97.23%	97.79%
Accuracy (testing)	95.39%	95.82%
Precision (macro avg)	95%	95%
Precision (weighted avg)	95%	**96**%
Recall (macro avg)	95%	95%
Recall (weighted avg)	95%	96%
F1 (macro avg)	95%	95%
F1 (weighted avg)	95%	96%

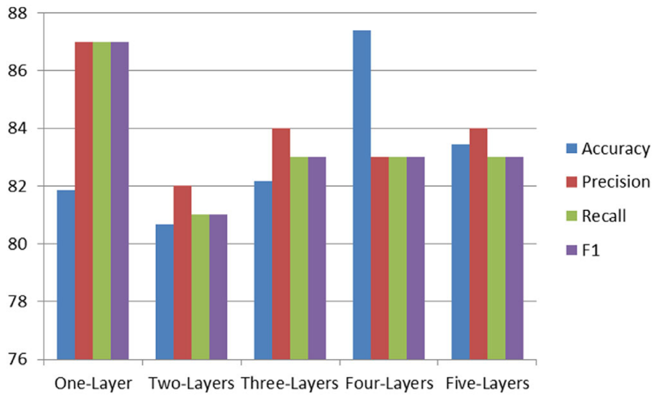

FIGURE 5 Stacked LSTM performance for German Credit for different number of hidden layers.

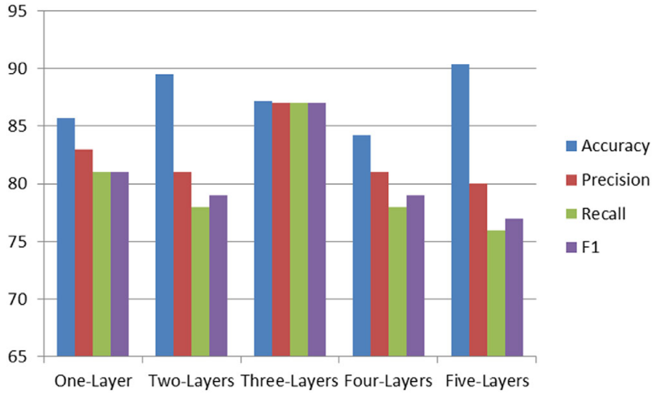

FIGURE 6 Stacked BDLSTM performance for German Credit for different number of hidden layers.

overall performance, and the aim was to setup the experiment for both networks with the same parameters. In Figure 5, the performance of Stacked LSTM model based on the number of layers for German Credit is presented, and Figure 6 Stacked BDLSTM.

For the German Credit dataset, three hidden layers were proved to be the best for the BDLSTM method, but outstanding results regarding Accuracy were evident for the four-layer LSTM German Credit model and five-layer BDLSTM model as well. However, other measures did not reach a satisfactory level, so a three-layer model would be the choice for the German Credit dataset.

For the Kaggle dataset, three-layer stacked LSTM is also the best choice considering the four most relevant measures Accuracy, Precision, F1 score, and Recall (Figures 7 and 8).

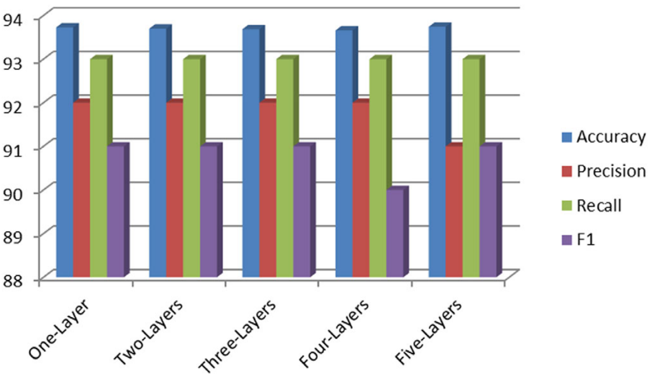

FIGURE 7 Stacked LSTM performance for Kaggle Credit for different number of hidden layers.

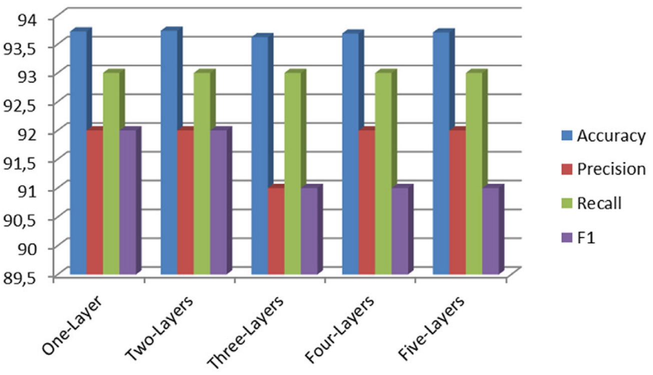

FIGURE 8 Stacked BDLSTM performance for Kaggle Credit for different number of hidden layers.

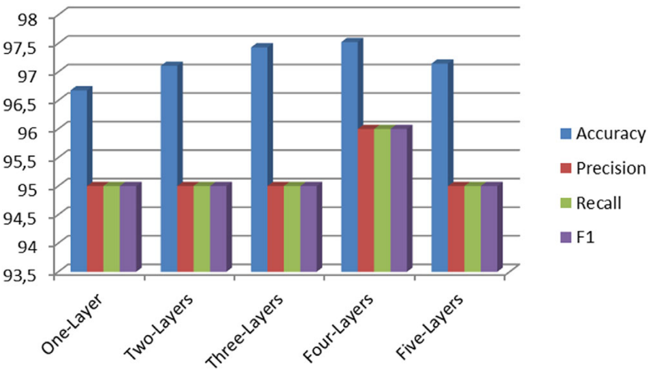

FIGURE 9 Stacked LSTM performance for Microcredit dataset for different number of hidden layers.

For Kaggle dataset, change in number of layers does not result in significant change in the model performance for both networks, and for Microcredit dataset the influence of number of layers are similar to the one described for German dataset since both of them have similar structure, but difference in number of instances and data quality (Figures 9 and 10).

5.3 | Comparison with previous studies

Comparisons to other state-of-the-art models show that proposed enhancements achieve outstanding results. Specific approaches to dataset fitting into sequence matrices and hyperparameter setup were applied in the three layer Stacked LSTM and BDLSTM network building. That is described in detail in Sections 4.2, 4.3, and 4.4.

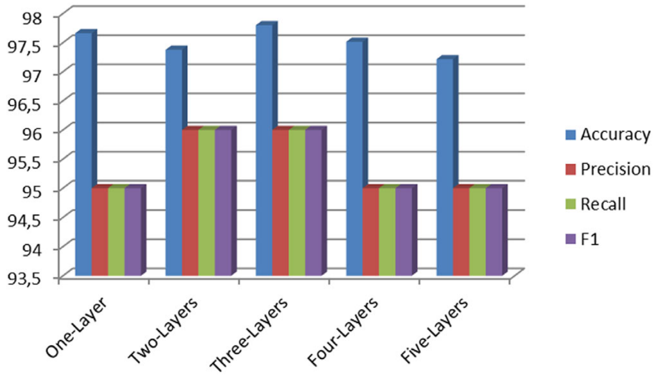

FIGURE 10 Stacked BDLSTM performance for Microcredit dataset for different number of hidden layers.

The outcomes divided by measures for both Deep Neural Networks show excellent results achieved without complex ensemble and without additional application of SMOTE or boosting algorithms. According to research performed by Xolani et al.[55] various credit scoring studies published different reports on PCC (Pearson correlation coefficient) and AUC on the same datasets due to data preprocessing. Accuracy reached a percentage of 87.19% for the training German credit dataset and 88.89% for the unseen out of sample test dataset (stacked BDLSTM). Even though in the study,[5] the basic LSTM method was additionally supported by SMOTE and AdaBoost, the published results were poorer compared to the results in this study.

The proposed stacked LSTM model achieved the highest accuracy of 93.72% for training and 93.53% for an unseen out-of-sample Kaggle test dataset. Stacked BDLSTM had an accuracy of 93.71% for training Stacked BDLST and 93.34% for the unseen Kaggle test sample dataset. There are only a few works that report accuracy as a measure. Publications such as[21] reported TPR (true positive rate), TNR (true negative rate), GMean (Geometric mean) and AUC measures, and Reference 56 reported AUC and Logloss.

Microcredit Credit model performances were compared to previous publication[38] and improved by applying our novel approach. Training accuracy applying Stacked BDLSTM reached 97.79%, compared to the previous 96.2% achieved by the ensemble of algorithms and SMOTE. What additionally proved our models' efficiency and generalizability was testing over an out-of-sample dataset where the results were excellent and without overfitting.

Table 8 shows our model performance compared to results achieved by recent algorithms' applications published in higher Impact Factor journals, divided by credit scoring datasets regarding accuracy if reported. Our models performances are marked with yellow.

6 | CONCLUSIONS

The main objective of this study was to improve results on classical publicly available credit scoring data using novel Deep Learning algorithms. We have adapted two Deep Neural Network architectures, which have not been applied before, to solve credit scoring problems and compare them to the results published in previous works. Generally, in the finance domain, the lack of available credit risk datasets is evident. Therefore, the real-world Microcredit dataset was also introduced, in addition to the public German and Kaggle datasets.

Proposed model enhancements largely depend on the setup of hyperparameters, but unique and novel dataset modeling with different principles in converting the data into sequence matrices was crucial. Although LSTM and BDLSTM based models are dominant for financial time series forecasting, the full advantages of the techniques in credit scoring cannot be exploited without proper dataset modeling. Stacked LSTM and BDLSTM showed better performance compared to other algorithms that have very complex architectures, extremely slow model building, excessive data transformation, additional feature selection, oversampling, and boosting. Feature selection and SMOTE are not applied in this study on purpose, to preserve simplicity and to demonstrate our model's supremacy. This means that although LSTM and BDLSTM based models are dominant for financial time series forecasting, only after transforming data into stationary allowed us to use all of the advantages of the methods also in credit scoring.

An important area for future work could be to focus more on the interpretability of the results and decisions made by complex black boxes. To overcome this drawback, methods such as LIME (local interpretable model-agnostic explanations), SHAP (shapley additive explanations), DALEX (model agnostic language for exploration and explanation), XAI (explainable artificial intelligence) methods and so forth could be utilized.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest.

DATA AVAILABILITY STATEMENT

a. German Credit data set that support the findings of this study are openly available in Reference 36: https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data). (Accessed 26 July 2021).

b. The Kaggle data set that support the findings of this study are openly available in Reference 37: https://www.kaggle.com/brycecf/give-me-some-credit-dataset (Accessed 26 July 2021).

c. Microcredit data set was used in the literature Gicic Subasi, 2018. Due to privacy and ethical concerns the data cannot be made available to the public. Due to confidentiality agreements, data can be made available subject to a nondisclosure agreement. https://doi.org/10.1111/exsy.12363 (Accessed 26 July 2021).

ORCID

Adaleta Gicić https://orcid.org/0000-0001-9114-1826

Dženana Đonko https://orcid.org/0000-0002-8933-006X

REFERENCES

1. Durand D. *Risk Elements in Consumer Installment Financing*. National Bureau of Economy Research; 1941:189-201.
2. Crook J, Edelman D, Thomas LC. Recent developments in consumer credit risk assessment. *Eur J Oper Res*. 2007;183(3):1447-1465.
3. Bank for International Settlements (BIS). Basel II: International convergence of capital measurement. *Basel Committee on Banking Supervision*. 2006;347.
4. GDPR, Reform of eu data protection rules, 2018 Accessed 17 September 2020. https://www.legislation.gov.uk.
5. Shen F, Zhao X, Kou G, Alsaadi F. A new deep learning ensemble credit risk evaluation model with an improved synthetic minority. *Appl Soft Comput J*. 2021;98:106852.
6. Ozbayoglu AM, Gudelek MU, Sezer OB. Deep learning for financial applications: a survey. *Appl Soft Comput*. 2020;93:106384.
7. Xia Y, Zhao J, He L, Li Y, Niu M. A novel tree-based dynamic heterogeneous ensemble method for credit scoring. *Expert Syst Appl*. 2020;159:113615.
8. Zhang W, He H, Zhang S. A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: an application in credit scoring. *Expert Syst Appl*. 2018;121(1):221-232.
9. Breeden JL. A survey of machine learning in credit risk. *Journal of Credit Risk*.2020; 17(3).
10. Guerra P, Castelli M. Machine learning applied to banking Supervision a literature review. *Risks*. 2021;9(7):136.
11. Laborda J, Ryoo S. Feature selection in a credit scoring model. *Mathematics*, License CC BY 4.0. 2021;9(7):746.
12. Lessmann S, Baesens B, Seow H-V, Thomas LC. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. *Eur J Oper Res*. 2015;247(1):124-136.
13. Huang S-C, Wu C-F, Chiou C-C, Lin M-C. Intelligent FinTech data mining by advanced deep learning approaches. *Comput Econ*. 2021;59:1407-1422.
14. Babaev D, Savchenko M, Tuzhilin A, Umerenkov D. ET-RNN: Applying deep learning to credit loan applications. *KDD 2019:* Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery&Data Mining. Association for Computing Machinery; 2019:2183-2190.
15. Zhang Y, Wang D, Chen Y, Shang H, Tian Q. Credit risk assessment based on long short-term memory model. *International Conference on Intelligent Computing, Intell Comput Theories Appl*. 2017; 10362:700-712.
16. Wang C, Han D, Liu Q, Luo S. A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism LSTM. *IEEE Access*. 2018;7:2161-2168.
17. Shubham M, Raina A, Singh SJ, Vashishtha A, Pandit AK. A Gaussian process-based approach toward credit risk modeling using stationary activations. *Concurr Comput Practice Exp*. 2021;34(5). doi:10.1002/cpe.6692.
18. Yeh S-H, Wang C-J, Tsai M-F. Deep belief networks for predicting corporate defaults. *24th Wireless and Optical Communication Conference (WOCC)*. IEEE; 2015.
19. Tran K, Duong T, Ho Q. Credit scoring model: a combination of genetic programming and deep learning. *Future Technologies Conference (FTC)*, San Francisco, CA, USA. 2016;145-49. doi:10.1109/FTC.2016.7821603.
20. Ramasamy S, Rajaraman K. A hybrid meta-cognitive restricted boltzmann machine classifier for credit scoring. *TENCON 2017–2017 IEEE Region 10 Conference*. IEEE; 2017. doi:10.1109/TENCON.2017.8228247
21. Tomczak JM, Zięba M. Classification restricted Boltzmann machine for comprehensible. *Expert Syst Appl*. 2015;42:1789-1796.
22. Neagoe V-E, Ciotec A-D, Cucu G-S. Deep convolutional neural networks versus multilayer perceptron. *Int Conf Commun*. 2018;201-206.
23. Zhang R, Zhiyi Q. Optimizing hyper-parameters of neural networks with swarm intelligence: a novel framework for credit scoring. *PLoS One*. 2020;15(6). doi:10.1371/journal.pone.0234254
24. Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. *Appl Soft Comput Jl*. 2020;90:106181.
25. Osei S, Mpinda BN, Sadefo-Kamdem J, Fadugba J. *Accuracies of some Learning or Scoring Models for Credit Risk Measurement*. HAL; 2021.
26. Zhu B, Yang W, Wang H, Yuan Y. A hybrid deep learning model for consumer credit scoring. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE; 2018:205-208.
27. Dastile X, Celik T. Explainable, making deep learning-based predictions for credit scoring. *IEEE Access*. 2021;9:50426-50440.
28. Zhou X, Zhang W, Jiang Y. Personal credit default prediction model based on convolution neural network. *Math Probl Eng*. 2020;2020:5608392.
29. Pławiak P, Abdar M, Pławiak J, Makarenkov V, Acharya R. DGHNL: a new deep genetic hierarchical network of learners for prediction of credit scoring. *Inf Sci*. 2020;516:401-418.
30. Tian J, Liu X, Li M. An incremental learning ensemble method for imbalanced credit scoring. *IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China*. IEEE; 2019: 754-759.
31. Zhang H, He H, Zhang W. Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. *Neurocomputing*. 2018;316(17):210-221.
32. Dumitrescu E-I, Hué S, Hurlin C, Tokpavi S. Machine learning or econometrics for credit scoring: Let's get the best of both worlds. *SSRN Electron J*. 2020.

33. Liu W, Fan H, Xia M. Step-wise multi-grained augmented gradient. *Eng Appl Artif Intell*. 2021;97:104036.
34. Camacho-Urriolagoitia O, López-Yáñez I, Villuendas-Rey Y, Camacho-Nieto O, Yáñez-Márquez C. Dynamic nearest neighbor: an improved machine learning classifier and its application in finances. *Appl Sci*. 2021;11(19):8884.
35. Yotsawat W, Wattuya P, Srivihok A. A novel method for credit scoring based on cost-sensitive neural network ensemble. *IEEE Access*. 2021;9:78521-78537.
36. Hofmann H. *UCI Machine Learning Repository: Statlog (German Credit Data) Data Set*. Institut fur Statistik und εOkonometrie Universitεat Hamburg; 1994.
37. Freshcorn B, Give Me Some Credit:: 2011 Competition Data, Accessed 26 July 2021. https://www.kaggle.com/brycecf/give-me-some-credit-dataset.
38. Gicić A, Subasi A. Credit scoring for a microcredit data set using the synthetic minority oversampling technique and ensemble classifiers. *Expert Syst*. 2019;36(2):e12363.
39. Shorten C, Khoshgoftaar T. A survey on image data augmentation for deep learning. *J Big Data*. 2019;6(60).
40. Sepp H, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.
41. Greff K, Srivastava RK. Lstm: a search space odyssey. *IEEE Trans Neural Networks Learn Syst*. 2016;28(10):2222-2232.
42. Xiang H, Chen B, Yang M, Xu S, Li Z. Improved direction-of-arrival estimation method based on LSTM neural networks with robustness to array imperfections. *Appl Intell*. 2021;51:4420-4433.
43. Cui Z, Ke R, Pu Z, Wang Y. Stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. *Transport Res C Emerg Technol*. 2018;118:102674.
44. Cui Z, Ke R, Pu Z, Wang Y. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Trans Res C Emerg Technol*. 2020;118:102674.
45. Schuster M, Paliwal K. Bidirectional recurrent neural networks. *IEEE Trans Signal Process*. 1997;45(11):2673-2681.
46. Li C, Bao Z, Li L, Zhao Z. Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multi-modal emotion recognition. *Inf Process Manag*. 2020;57(3):102-185.
47. Onan A, Toçoğlu MA. A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification. *IEEE Access*. 2021;9:7701-7722.
48. Caruana R, Alexandru N-M. An empirical comparison of supervised learning algorithms. *ICM L' 06:Proceedings of the 23rd International Conference on Machine Learning*. 2006.
49. Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Mach Learn*. 2011;85(3):333-359.
50. Graves A, Mohamed A-R, Hinton G. Speech Recognition with Deep Recurrent Neural Networks, *International Conference on 38th Acoustics, Speech, and Signal Processing* 1988. *ICASSP-88*; 2013.
51. Sutskever I, Martens J, Dahl G, Hinton G. On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on Machine Learning*, Atlanta GA USA, 2013.
52. Sarker IH. Machine learning: algorithms, real-world applications and research directions. *SN Comput Sci*. 2021;2(3):160.
53. Wang F-P. Research on application of big data in internet financial credit investigation based on improved GA-BP neural network. *Hindawi, Complexity*. 2018;2018:7616537.
54. Chris A. *Machine Learning with Python Cookbook by Chris Albon*. O'Reilly Media, Inc.; 2018.
55. Xolani D, Celik T, Potsane M. Statistical and machine learning models in credit scoring: a systematic literature survey. *Appl Soft Comput Jl*. 2020; 91:106263.
56. Guo Y, He J, Xu L, Liu W. A novel multi-objective particle swarm optimization for comprehensible credit scoring. *Soft Comput*. 2019;23(18):9009-9023.

How to cite this article: Gicić A, Đonko D, Subasi A. Intelligent credit scoring using deep learning methods. *Concurrency Computat Pract Exper*. 2023;35(9):e7637. doi: 10.1002/cpe.7637