DESIGNING A DEEP LEARNING-BASED FINANCIAL DECISION SUPPORT SYSTEM FOR FINTECH TO SUPPORT CORPORATE CUSTOMER'S CREDIT EXTENSION

Arodh Lal Karn[1], V. Sachin[2], Sudhakar Sengan[3], Indra Gandhi V[4], Logesh Ravi[5], Dilip Kumar Sharma[6], Subramaniyaswamy V[7*]

[1]School of Management, Northwestern Polytechnical University, Xian, Shaanxi-710072, China

[2]Member Technical Staff, Zoho Corporation, Chennai, Tamil Nadu, India

[3]Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu, India

[4]School of Electrical Engineering, Vellore Institute of Technology, Vellore, India

[5]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science & Technology, Avadi, Chennai, India

[6]Department of Mathematics, Jaypee University of Engineering and Technology, Guna, Madhya Pradesh, India

[7]School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

Email: alkmaithili@nwpu.edu.cn[1], velsachin98@gmail.com[2], sudhasengan@gmail.com[3], arunindra08@gmail.com[4], LogeshPhD@gmail.com[5], dilipsharmajiet@gmail.com[6], vsubramaniyaswamy@gmail.com[7]*(corresponding author)

DOI: https://doi.org/10.22452/mjcs.sp2022no1.9

ABSTRACT

In the banking business, Machine Learning (ML) is critical for averting financial losses. Credit risk evaluation is perhaps the most important prediction task that may result in billions of dollars in damages each year (i.e., the risk of default on debt). Gradient Boosted Decision Tree (GBDT) models are now responsible for a large portion of the improvements in ML for predicting credit risk. However, these improvements begin to stagnate without adding pricey new data sources or carefully designed features. In this work, we describe our efforts to develop a unique Deep Learning (DL)-based technique for assessing credit risk that does not rely on additional model inputs. We present a new credit decision support approach with Gated Recurrent Unit (GRU) and Convolutional Neural Networks (CNN) that uses lengthy historical sequences of financial data while requiring few resources. We show that our DL technique, which uses Term Frequency-Inverse Document Frequency (TF-IDF) pre-classifiers, outperforms the benchmark models, resulting in considerable cost savings and early credit risk identification. We also show how our method may be utilized in a production setting, where our sampling methodology allows sequences to be effectively kept in memory and used for quick online learning and inference.

Keywords: Financial Decision Support System, Term Frequency, Machine Learning, Convolutional Neural Networks

1.0. INTRODUCTION

The banking sector mainly functions on the persistent mobility of funds. There are about a million transactions from its million accounts in thousand branches on an average in a day. The amalgamation of these data on a large scale can create a huge dataset for the bank to evaluate in order to get an understanding. Hence, the necessity arises for the banks to install a device for converting these data into information for which there should be consistency between the heterogeneous data sources. Here, the role of the Decision Support System (DSS) becomes instrumental [1].

The process of data accumulation and evaluation performed by DSS generates comprehensive information reports. So, there is a substantial variation between DSS and an application of usual operations that aims to gather data rather than evaluate it. The planning departments in a firm use a DSS in the operations in which the managers use the data for Decision-Making Process (DMP) collected by the DSS and generate a report. For inventory and operations-based data, sales forecast, and updating of information to customers in a comprehensible manner, we use DSS. DSS can be a hypothetical application in several knowledgeable realms, from a firm to forest management and the medical domain. Real-time reporting is one of the significant domains in which DSS is used enormously. Also, the organizations that participate in Just-In-Time (JIT) inventory management make use of it on a large scale. Orders are placed "just in time" in a JIT inventory system to prevent production delays and negative domino effect for which the organization needs real-time data on stock levels. As a result, when compared with a traditional system, the DSS is more tailored to the individual or organization's decision-making process [2].

1.1. Types of Decision Support Systems

There are five types of DSSs [3]:

a) **Communication-Driven DSS (D-DDSS):** Communication-Driven DSS is mainly determined to facilitate users to integrate technology in a web/client-server.

b) **Data-Driven DSS:** The managers, employees, and suppliers of products are aimed at by D-DDSSs. In order to question a database to find solutions for particular problems, these D-DDSSs are applied. Through mainframe systems, client or server links/web is applied.

c) **Document-Driven DSS:** The broad user base is aimed at document-driven DSSs. The unique set of keywords or search items are used to seek web pages and locate documents. The web/client or server system facilitates the installation of D-DDSSs.

d) **Knowledge-Driven DSS (K-DDSS):** K-DDSSs occupy a wide range of systems and organizational users besides the other interactors with the organization. The primary aim of this system is to give management-related advice. Client/server systems, web or software operated on stand-alone Personal Computers (PCs) are some technologies that enable the installation.

e) **Model-Driven DSS (M-DDSS):** According to the decision on how the model is installed for scheduling, decision making, and so on, managers or organizational members make use of M-DDSSs. Through software, hardware, client/server systems, or web, these can be applied.

1.2. Benefits of DSS

DSS is indeed very significant provided that the well-planned DMP is key to the bank's functioning. The management of customer queries and cutting of operating costs are facilitated by DMP, and the competence and effectiveness of DMP are enhanced in the proper implementation of the system.

The following are some of the primary benefits of DSS: i) Rapid problem-solving and data processing, ii) Risk management and predicting the future, iii) Collection and analysis of proof to support a decision, iv) Improved quality of decision-making process, and v) Capability of feeding learned lessons back into the D-MP again.

The essentials of D-MP like determination, decision, and set of actions are supported by a DSS, an information system powered by humans, automated, or an amalgamation of both. The voluminous data of unstructured nature are analyzed, and information that enables problem-solving and aid in D-MP are gathered so that the information system helps the middle and high-level management.

1.3. DSS's Components

A DSS framework contains the following three major components:

a. **Model Management System (MMS):** Managers can utilize the MMS models in their DMP. The DMP specific for organization's financial status and the need for a prediction about a product or service are used in the models.

b. **User Interface (UI):** The end-user of a DSS assisted by the tools is included in the UI for navigating the system.

c. **Knowledge Base (KB):** Information collected from internal sources like the transaction process system and external sources like newspapers and online databases are included in the KB.

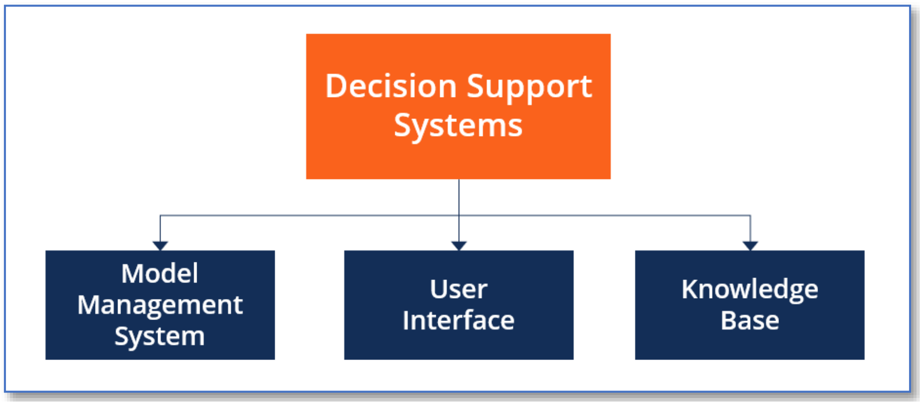

Fig. 1: Component of decision support system

An accurate and automated financial decision (Fig. 1)is taken using the heterogeneous prospects provided by Financial Technologies (FinTech) that give unique solutions to Financial Institutions (FI). The American Bankers Association's recent survey stated that digital loan origination channels are used by 50% of huge banks in the viewpoint of a lending value chain. But in the primary process like loan application, uploading of a document, including e-Signatures, and making the corporate customer communication digitalized, most of the digitization takes place. Predominantly, the decision process of the downstream credit is manual, with just 13% and 32 % of smaller and larger banks, respectively, that provide automated and immediate loan approvals [4].

The customers' non-gratification about the conventional FI is the cause of this low adoption of automated DMP. The Fed Survey conducted in 2017 revealed that 33% of corporate loan applicants were unhappy about the credit decision that took a longer waiting time, and 28% of the respondents complained about the challenging process of the application. At the same time, the dissatisfaction of online lenders for these reasons is only 10% [5].

The following are the main problems encountered by FI's credit decisioning using traditional approaches: (i) Heterogeneous, disparate systems containing categorized data, (ii) data related to the customer for the products used, (iii) Strict legacy systems, (iv) Absence of transparency and existence of bias. Because of multiple challenges like the ability to take the risk by incorporating innovative technologies, FI has reliability still on manual efforts though there is a desire in automating the credit DMP. Moreover, the strategic implementation that causes a change in rules in the DMP takes more time because of restricted automation, and this takes more time to realize the benefits for FI. Consequently, the competitive benefits are lost by FI in the market.

If FI has relied upon conventional credit DMP, then it will be unable to develop and meet the expectations of the customer. The majority of shares from the portfolios of the bank is already captured by FinTech. According to Transunion, in 2018, 38% of the total personal loans were issued by FinTech, whereas 5 years ago, the share of traditional banks has come down from 40% to 28%. Over the past few years, numerous venture capital deal activities in the FinTech domain facilitated inventive ideas via End-to-End technology-driven credit lending transformation.

From the process of data collection until determining the approval or rejection of a customer's credit application, FIs are supposed to start analyzing their conventional credit DMP. The opportunities for the assimilation of automation through the capabilities of FinTech's recent technology like Artificial Intelligence (AI)/Machine Learning (ML) are uncovered by this process.

The paper is organized as follows: Section 2.0 presents the literature review for the current scope followed by preliminary concepts employed in this work presented in Section 3.0, Section 4.0 presents the methodology, and Section 5.0 and Section 6.0 present the experimental results and conclusion.

2.0 LITERATURE REVIEW

A long time ago, models were developed to assess the similarity of company default by governments, fund managers, money lenders, FI, and financial market players. Statistical models are relied upon by traditional approaches. There could be a classification of observations into good and bad payers using the score generated by Altman. After this works, the model is explicitly designed for sectors and segmentation using the other applications that have been developed [6], [7], [8], [9]. As opposed to Altman, for default estimation, Logistic Regression (LR) analysis was applied by Ohlson [10], one of the first researchers. The default probability of a potential borrower determines Ohlson's model. The performance of discriminant analysis and LR and various consequent studies have attempted to carry out similar tests. The massive time-series data cannot be used automatically to assimilate with the discriminant analysis and LR and depend on the standard mean-value theory.

In the ML domain, the pattern recognition techniques used by the studies have been dynamically developed to overcome the setbacks of statistical models by representing the outstanding performance of ML models over conventional classification methods. The AI systems like Genetic Algorithms (GA) and Neural Networks (NN) are used by some of these works [11], [12], [13]. The new works have demonstrated the Ensemble Models (EM) [14], [15].

Converting an open-access dataset into resources for urban planning and development is described by Milusheva et al. [16]. The automatic text classification of crisis-related communications is performed using the Artificial Intelligence for Disaster Response (AIDR) platform proposed by [17]. The messages posted by people during calamities are categorized into a set of user-defined information.

Moreover, in real-time or with low latency, it is mandatory that the entire process consume, process, and generate authentic information alone [18]. Though there are different degrees in the prediction accuracies of multiple studies, they are available in the literature, which makes an analogy between several AI/ML methods, when the essential financial variables [19] are used for developing the association between input variables and stock price [20], the Precision and dependability of the model increases. The well-accepted ML models like Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Decision Tree (DT) techniques like Random Forests (RF) and Model Trees (MT) in which the inputs contain technical indicators that are the basis of literature existing in the AI/ML applications for prediction of stock price [21]. Predicting stock price involves comparing numerous ML techniques that signify the complex non-linearity in the stock prices that LR models cannot capture. Moreover, when essential financial variables are selected as the model inputs, the ANNs and DT perform better [22].

The raw text is processed sequentially by the Recurrent Neural Network (RNN)[23]. A natural cycle is formed using the association between NN that permits information passing from one word to the next. Intensive learning of context-sensitive characteristics is made possible for RNN using this. RNN has limitations like the disappearance of gradient problem and short context reliability that frequently forbid its usage in real-time issues [24]. The Long Short-Term Memory (LSTM) that enables the process of successive inputs that lies mainly between correlated input signals [25] represents the enhancement of classical RNN.

The forget gates, which thwart the blasting gradients at the time of back-propagation and hence numerical inconsistencies, are utilized by LSTM for this purpose. Consequently, in multiple research areas, LSTMs have the most promising one, and hence the application of DL architecture is fruitful in our study on financial DSS.

3.0 METHODS AND MATERIALS

In order to anticipate the credit line extension depending on the financial disclosures, our methodology and the dataset are introduced in this section. Precisely, the bag-of-words is used with new techniques (Fig. 2 and Fig. 3) to compare naïve ML.

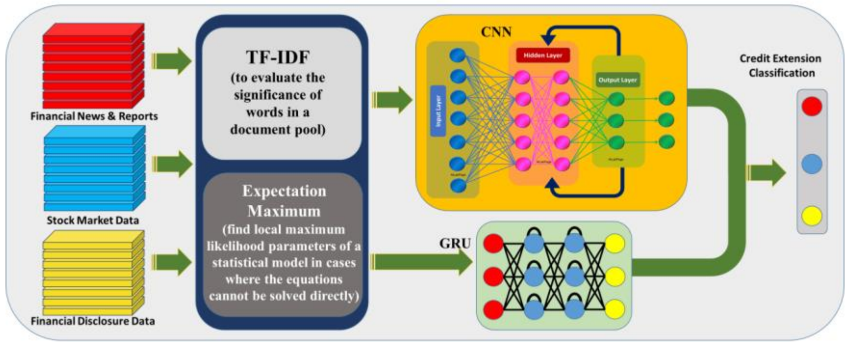

Fig. 2: The proposed financial DSS architecture

The face financials of every financial statement are the information source for the Financial Statement Data Sets (FSDS) that supplied information for this research study. From the exhibits to corporate financial reports stored in the Commission, these data are mined with the help of eXtensible Business Reporting Language (XBRL). The FSDS Data Sets are more compact when compared to the more extensive Financial Statement and Note Data Sets that offer numeric and narrative disclosures from the entire financial statements. Each registrant's financial reports are presented as such the same as they have filed earlier. Users can study and compare corporate disclosure information over time and between registrants by viewing the data in a flattened manner. Additional fields, such as a company's Standard Industrial Classification, are included in the data sets to make it easier to utilise the information.

The disclosures of petty stock companies that yield meager annual returns are ignored to minimize noise. Additionally, Ltd companies' publication of disclosures is alone selected, and a sample of 12,725 is yielded by this. By using a market model, the abnormal returns statement is calculated with Stock Market Index (SMI) data. Whereas during the days before disclosure, the Stock Market is devised through the CDAX. Based on the subsequent return, each disclosure encoded as "1" is labelled as positive or "0" as negative in the classification task.

The descriptive statistics are shown in Table 1. On average, 169.80 words are contained in Ad hoc releases that are extended more than the document utilization by the DL majority in its applications in the domain of Natural Language Processing (NLP). Our dataset comprises 12,444 different words. The dissemination of abnormal and nominal returns that are right-skewed are 25% and 75%, respectively. Due to various reasons that include mergers and acquisitions and also bankruptcy, our dataset consists of certain non-conformities of enormous magnitude. Since there can neither be recognition or filtering of values by a DSS in a live application, we intentionally include them in our data sample. A fact regarding the observation of an unbalanced set of labels is to be adapted in performance measurements. The positive label's appearance is 9% more than a label marked as negative in the case of nominal return.

Table 1: Stock market statistics summary

Variable	Observation of An Unbalanced Set	μ	σ	CV	Percentiles				
					10%	30%	60%	80%	98%
Incredible Return	12,725	0.679	6.967	9.891	-7.718	- 1.819	0.2817	2.839	10.12
Return on Investment Nominal	12,725	0.787	6.873	8.892	-7.617	1.617	0.2617	2.827	9.9718
Length (in Words)	12,725	169.1871	118.190	0.6919	62.102	96.28	138.1983	205.71	386.103

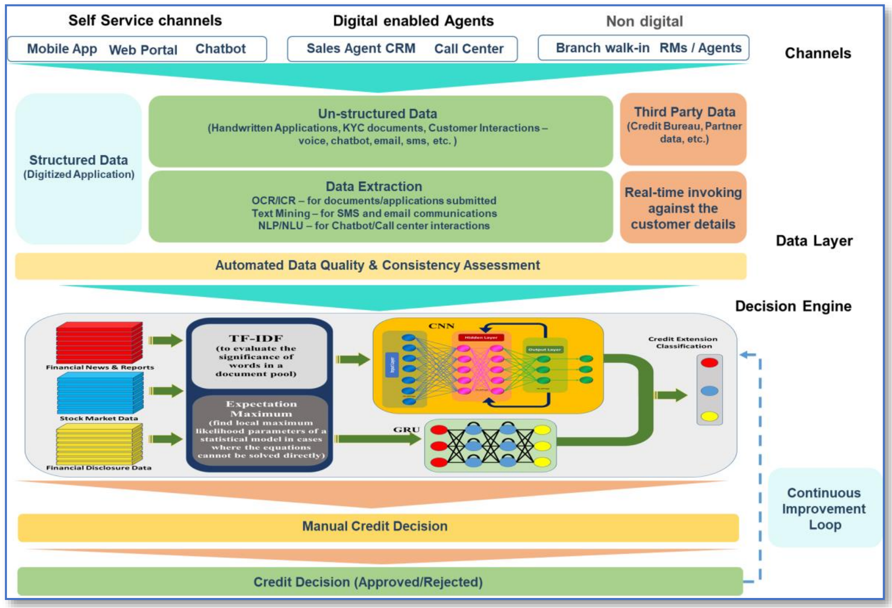

Fig. 3: Employing the architecture in a credit decision

The dataset is segmented as training and a test set for measuring the predictive performance. Our training datasets are provided by the first 80% of the time frame, whereas the final 20% defines our test datasets. The ignorance of the chronological order of disclosures in the latter process is a setback. Thus, data samples that are accessible by only experts get benefited from the training. The training and test sets are thus split in sequential order. As described later, there is an application of the related methodology in cross-validation. The dataset consists of disclosures of about 8,716 and 2,179 for training and testing, respectively.

3.1. Term Frequency-Inverse Document Frequency (TF-IDF)

To count a word in documents, the statistical model known as the TF-IDF technique is used. Each word has usually computed a weight that indicates the significance of the word in the document and corpus. In Information Retrieval and Text Mining, this technique is extensively used. In a document pool, the significance of the word is evaluated using TF-IDF. By multiplying two metrics, it may be calculated as 2-D matrix form of TF-matrix and 1-D matrix form of IDF. Various researchers proposed several refinements for the enhancement of the classical TF-IDF model, Eq. (1)

$$\mathrm{TF(I,j)} = \frac{\mathrm{n(I,j)}}{\mathrm{n(1,i)+n(2,j)+\cdots+n(q,j)}} \tag{1}$$

Suppose an amount of {D_1, D_2, …, D_N} with N documents is proposed. Let for $1 \le i \le p$. Let D_j= {t_1, t_2, …, t_q}, and any arbitrary document be a document comprising 'q' terms. Let an arbitrary feature word be 't_i'. *TF(i, j)* represents the term frequency of 't_i; relating to 'D_j' that may be defined by the equation if the number of appearances of 't_i' in document 'D_j' is denoted by *n(i,j)*.

Let the total number of documents be denoted by 'n_i' where 'ti' appears at least once. Then, IDF_i that represents TF-IDFof 't_i' related to the corpus may be defined as Eq. (2)

$$IDF_i = \log\left(\frac{N}{ni}\right)\mathrm{IDF(i)} \tag{2}$$

The ratio of the total number of documents available in hand with the number of documents where there is a presence of feature words is represented by IDF(i). In the entire quantity, this shows the dissemination of 't_i'. The assessment of the relevance of a specific word associated with the document is determined by TF-IDF, which is a Vector Space Model (VSM). Let hundreds of words be contained in a document. Let 5 be the frequency of the specific word "*research*". Then

$^{5}/_{100}$ = 0.05 will be the TF. Imagine that there are 40000 relevant documents and 400 times the words "research" appears among them. Then the calculation of IDF is $^{40000}/_{400}$=100. Hence, the value of TF-IDF=5 can be found.

A. The EM Algorithm in Its Classical Form: First, we presume that the design data is Z=(X, Y), but only X is demonstrated. In the case of Z=(X, Y), l(X, Y) provides a detailed log-likelihood and is the random variable vector for which we are looking for the MLE, Eq. (3)

$$\begin{aligned} Q(\theta;\theta_{\text{old}}) \quad &:= \mathrm{E}[l(\theta;\mathcal{X},\mathcal{Y}) \mid \mathcal{X},\theta_{\text{old}}] \\ &= \int l(\theta;\mathcal{X},y)p(y \mid \mathcal{X},\theta_{\text{old}})dy \end{aligned} \tag{3}$$

B. E-Step: The anticipated value of l(θ; X, Y) is computed by the E-Step of the EM algorithm provided with the observed data X, and θ_{old} as the existing parameter estimate. Notably, in the observed data X, p(θ j X; θ_{old}) is the conditional density of Y, and it is assumed that $\theta = \theta_{\text{old}}$, Eq. (4).

$$\theta_{new} := \max_{\theta} Q(\theta;\theta_{old}) \tag{4}$$

C. M-Step: Maximizing over θ in the M-step, in (1), the expectation is computed. Later we set $\theta_{\text{old}} = \theta_{\text{new}}$. Until the convergence of the sequence θ_{new}, the repetition of the two steps is wanted. In case of convergence, a local maximum is assured, and the reason behind the case is explained below. EM algorithm should be run multiple times with the help of various starting values of θ_{old} on every occasion if the log-likelihood function is suspected of having contained several local maximums. The best of the set of local maximums acquired from the different runs of the EM algorithm is deemed to be the ML estimate of θ.

To indicate a generic conditional PDF, we use p(j). As the In function is concave, now it is observed that (2) follows from Jensen's Inequality. It is explicit that because the term within the expectation becomes a constant if we take $\theta=\theta_{\text{old}}$, the inequality in (2) becomes equality. When $\theta = \theta_{\text{old}}$, we will thus have for all θ with inequality by letting the right-hand-side of (3) denoted by g(θ j θ_{old}), Eq. (5), Eq. (6) and Eq. (7).

$$\begin{aligned} l(\theta;\mathcal{X}) = \ln\, p(\mathcal{X} \mid \theta) \quad &= \ln \int p(\mathcal{X},y \mid \theta)dy \\ &= \ln \int \frac{p(\mathcal{X},y|\theta)}{p(y|\mathcal{X},\theta_{\text{old}})} p(y \mid \mathcal{X},\theta_{\text{old}})dy \end{aligned} \tag{5}$$

$$\begin{aligned} &= \ln\, \mathrm{E}\left[\frac{p(\mathcal{X},\mathcal{Y} \mid \theta)}{p(\mathcal{Y} \mid \mathcal{X},\theta_{\text{old}})} \mid \mathcal{X},\theta_{\text{old}}\right] \\ &\geq \mathrm{E}\left[\ln\left(\frac{p(\mathcal{X},\mathcal{Y}|\theta)}{p(\mathcal{Y}|\mathcal{X},\theta_{\text{old}})}\right) \mid \mathcal{X},\theta_{\text{old}}\right] \end{aligned} \tag{6}$$

$$\begin{aligned} &= \mathrm{E}[\ln\, p(\mathcal{X},\mathcal{Y} \mid \theta) \mid \mathcal{X},\theta_{\text{old}}] - \mathrm{E}[\ln\, p(\mathcal{Y} \mid \mathcal{X},\theta_{\text{old}}) \mid \mathcal{X},\theta_{\text{old}}] \\ &= Q(\theta;\theta_{\text{old}}) - \mathrm{E}[\ln\, p(\mathcal{Y} \mid \mathcal{X},\theta_{\text{old}}) \mid \mathcal{X},\theta_{\text{old}}] \end{aligned} \tag{7}$$

Hence, there must be an increase of l(θ; X) beyond l(θ_{old}; X) when the value of increases g(θ j θ_{old}) beyond g(θ_{old} j θ_{old}). By increasing Q(θ; θ_{old}) beyond θ, which is equal to increasing g(θ j θ_{old}) beyond θ, such a θ is found by M-step. The multiple application of the Q(θ; θ_{old}) function worth mentioning here will be a convex function of θ and thus make the optimization easier, Eq. (8).

$$l(\theta;\mathcal{X}) \geq g(\theta \mid \theta_{\text{old}}) \tag{8}$$

4. THE GRU-CNN HYBRID NEURAL NETWORK MODEL

4.1. Gated Recurrent Unit (GRU) Module

The Recurrent Neural Network (RNN) is a part of Artificial Neural Network (ANN) relevant for assessing and processing data related to time sequence different from CNN that depend on the mass between the layers. The preceding information from the preceding moment is preserved using the application of concealed layers by RNN, and the existing conditions and memories are affected by the output. In Fig. 4, the unrolled structure of RNN is shown for a comprehensive insight of RNN.

At which time *t*, input, output are represented by $x^{\langle t\rangle}$,$y^{\langle t\rangle}$and the respective weight matrixes of concealed layers, the input and weight matrixes of input and the output are represented by $\omega_{aa}^{\langle t\rangle}$, $\omega_{ax}^{\langle 1\rangle}$, and $\omega_{ay}^{\langle 1\rangle}$. The following formulae are given in Fig. 4, where one single concealed output layer of the bias vector is represented by $h^{<t>}$, and W represents the nonlinear activation function. When the output is near its correlated inputs, the performance of RNN is better; but when a long time interval and a large number of weights exist, the output caused by the gradient vanishing issue will have a minor impact on the input. The issues involved in RNN's gradient vanishing and simple concealed layer structure are solved using an exclusive kind of RNN called GRU.

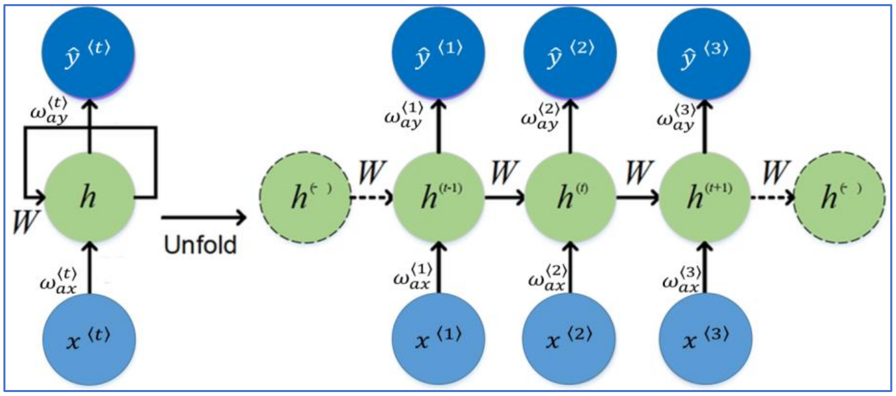

Fig. 4: RNN structure

The Gated Recurrent Neural Network (GRNN) structure is integrated with GRU, a variant of LSTM, and in GRU, two gates exist viz., Reset and Update gate and three Gates viz., Input, Output and Forgetting gate when comparing with LSTM. In the meantime, as the convergence of GRU is quicker than LSTM during training, there are lesser training parameters in GRU than LSTM.

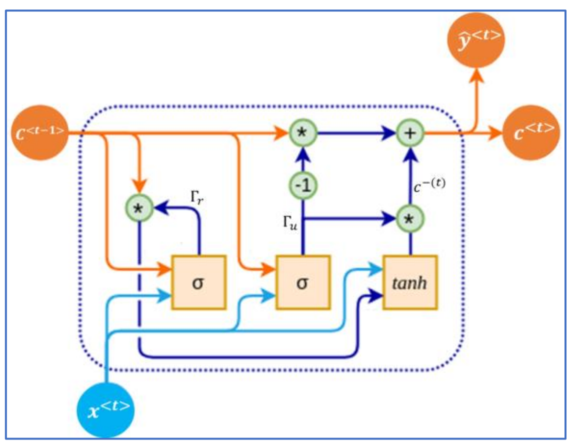

Fig. 5: GRU structure

The GRU structure is shown in the above Fig. 5, where the activation functions are σ and tanh, and the existing unit's input is $c^{\langle t-1\rangle}$ that is also the preceding unit's output; the existing unit's output is $c^{\langle t\rangle}$ that connects the subsequent unit's input. The training data's input is $x^{\langle t\rangle}$.This unit's outcome $y^{\langle t\rangle}$ is created by the activation function, the reset and the update gate are represented by Γ_r and Γ_u respectively, and the computation of candidate activation $c^{\langle t\rangle}$ is identical to that of the conventional reiterant unit. Here in GRU, there are two gates, and the update gate preserves preceding information to the existing state; Γ_u's ***.e*** value ranges from 0 to 1, the retaining of more preceding information than the closer Γ_u is to '*0*'; the other is the reset gate to ease whether there should be an amalgamation of the existing status and preceding information. Γ_r's **.** The value ranges from -1 to 1; the preceding information is ignored much if the value of Γ_r is higher. The following is GRU's formula:

where the update and reset data's training weight matrix, and the candidate's activation $c^{\langle t \rangle}$ are represented by ω_u, ω_r, and ω_c, respectively, and the bias vectors are b_u, b_r, and b_c, Eq. (9).

$$\begin{aligned}\Gamma_u &= \sigma\left(\omega_u\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right),\\ \Gamma_r &= \sigma\left(\omega_r\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right),\\ \tilde{c}^{\langle t \rangle} &= \tanh\left(\omega_c\left[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)\\ c^{\langle t \rangle} &= (1-\Gamma_u) * c^{\langle t-1 \rangle} + T_u * \tilde{c}^{\langle t \rangle}\end{aligned} \tag{9}$$

4.2. Convolutional Neural Network (CNN)

A kind of ANN is known as CNN that is capable of processing high-dimensional data. In video recognition, visual image, and text categorization, this application is expected. The Spatiotemporal Matrix (SM) is designed on the basis of the time sequence of the registrant's return fillings. The display of SM is shown as Eq. (10).

$$X = \begin{bmatrix} X_1(1) & X_1(2) & \cdots & X_1(n) \\ x_2(1) & x_2(2) & \cdots & x_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ X_1(1) & X_k(2) & \cdots & X_k(n) \end{bmatrix} \tag{10}$$

The k^{th} registrant filings are represented by k, the n^{th} time sequence is represented by n, and the data recorded by the k^{th} at n time is represented by $X_k(n)$. The SM is processed using CNN for extracting the feature from the SM. Fig. 6 shows the structure of CNN. At first, numerous 2-D SMs are stored into 3-D matrix blocks, and later with a convolution operation, these blocks were deployed. A CNN operation aims to obtain a highly abstract feature, and its outputs are applied with a pooling operation once the operation is completed. However, the size of matrixes and the nodes' count can be reduced by the pooling operation for having reduced parameters in the whole NN though it does not modify the depth of the input matrix. The highly abstract feature was acquired and flattened to a 1-D vector in order to be linked with the Absolutely Connected Layer (ACL) after recurrent convolution and pooling operations. Later, the calculation of the weights and bias parameters of the ACL are performed recurrently. Ultimately, through the output of the activation function, the anticipation outcomes are acquired.

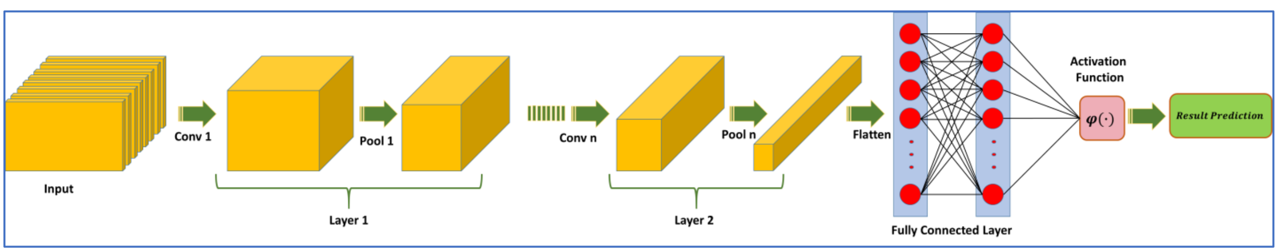

Fig. 6: CNN structure

4.3. GRU-CNN Hybrid Neural Networks (GR-CNN)

A hybrid neural network that combines the GRU module's ability to process time sequence data with the CNN module's ability to handle high-dimensional data was developed, and the construction of GRU-CNN hybrid neural networks is depicted in Figure 7. The proposed GRU-CNN hybrid neural network system is composed of a GRU module and a CNN module. The CNN module is particularly adept at processing 2-D input, such as spatiotemporal matrices and pictures.

The SMs data provide the local features directly using the local connection and shared weights by the CNN module, and through the convolution and pooling layer, the effective representation is acquired. Two convolution layers and a flatten operation are contained in the structure of the CNN module, and a convolution and pooling operation are contained in each convolution layer. The high-dimensional data are flattened into 1-D data, and with the ACL, the outputs of the CNN module are connected after the second pooling operation. Alternatively, capturing the long-term dependency is the goal of the GRU module, and in the historical data, the GRU module can learn valuable information for a prolonged time via the memory cell, and the forget gate will forget the invaluable information. The time sequence data are the inputs of the GRU module; many gated reiterant units are contained in the GRU module, and with the ACL, these gate reiterant units' outputs are connected. At last, calculating the mean value of all neurons in the ACLs gives the load predicting results.

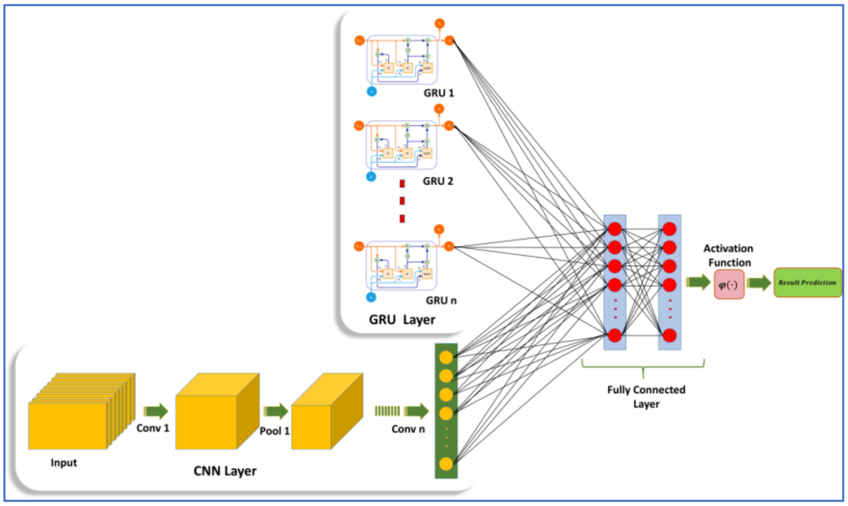

Fig. 7: GRU-CNN HNN model

5.0 PARAMETER TUNING

To detect the outstanding parameters based on the time-series cross-validation, which deploys a continuous prediction, algorithm 1 depicts the tuning. The $\mathbb{T}$training data is divided into k disjoint subsets$\mathbb{T}_1, \ldots, \mathbb{T}_k$ that are sequenced chronologically. Then, the tuning range is set for each parameter, repeated on all feasible amalgamations and subsets of $\mathbb{T}$. On the validation set$\mathbb{T}_i$, the effective performance $eff_i$of the approach is measured in each recurrence. Whereas the points in time that precedes act as training, the subsets $\mathbb{T}_1, \ldots, \mathbb{T}_{i-1}$are used. Eventually, the ideal parameter setting is returned.

5.1. Algorithm for Hyper Parameter Tuning

Input:

$\mathbb{T}$, the chronologically ordered training data

$\mathbb{S} = \{ \mathbb{T}_1, \mathbb{T}_2, \ldots \mathbb{T}_s)$, contains disjoint subsets of training data

Tuning parameter range, $\mathbb{R}_n$; n = 1,2,…,m

Repeat

Repeat

Using the Data from Subset $\mathbb{T}_1, \mathbb{T}_2, \ldots \mathbb{T}_j$, Train Model $\mathcal{M}$;

$eff_j \leftarrow the\ Effective\ Performance\ of\ Model\ \mathcal{M}\ on\ \mathbb{T}_j$

$j \leftarrow j + 2;$

Until$j < s;$

$$eff_{(\wp_1,\wp_2,\ldots,\wp_m)} \leftarrow \frac{1}{s}\sum_{j}^{s} eff_j$$

Until $(\wp_1, \wp_2, \ldots, \wp_m) in\ \mathbb{R}_1 \times \mathbb{R}_2 \times \cdots \times \mathbb{R}_m$

Return $arg \max_{(\wp_1,\wp_2,\ldots,\wp_m)} eff_{(\wp_1,\wp_2,\ldots,\wp_m)}$

In the online appendix, the tuning parameters are described. A grid search is performed on the training dataset to tun the baseline techniques' whole parameters and a 10-fold time-series cross-validation. Because of high computational demand, the last 15% of the training set for validation is used for tuning the parameters of architectures in the case of DL. The learning rates centric Adam algorithm is tuning on the interval [0.00009; 0.009], and0.0005 of step size is utilized to optimize the Deep-NN, whereas the Kaiming algorithm initializes weights. On the {25, 35,…, 100} set, the tuning of the word embeddings' size is performed. The persistent even distribution U(-0.1; 0.1) is the base of each NN layer's size located inside the CNN and GRU to the word embeddings' dimension, and the word embeddings are initialized.

5.1. Computational Results

The outcomes of our model that is depicted in Section 4 have been discussed in this section. The results of Son et al., 2019; Chou et al., 2017 and Figini et al., 2017 are compared with our results. The following metrics are used to measure the performance: Matt Coefficient, Precision and Recall, AUC, F1-score, and Log-Loss. We do not add the definition of performance measure in order to have Precision. Matthews (1975) and Japkowicz and Shah (2011) can be referred by the interested reader. We identified a suitable threshold of 60% regarding the II-Phase data set creation. Indeed, the companies performing well can be deemed active in another 1 to 5 years and can be maintained to not bias our data set toward bankrupted companies.

Each year, the companies that become bankrupt ought to be deemed as approximately 3% theoretically; and this fact could be confirmed by a 60% threshold. Table 2 presents the application of various ML methods with the results of the ML method where standard sampling's results are presented by "1st Round", whereas the performance of the II-phase technique is presented in the "2nd Round". Just the results are presented with the help of the best threshold (60%) for easiness. Our II-phase process achieved better performance than standard stratification, as understood from the results.

The [26], [27], [28] models are outperformed by the proposed algorithm in our work, whereas a similar performance is achieved but with a slight betterment compared to Son et al. (2019). When compared to the other models (Log Loss: Son et al., =0.2895, Chou et al., (2017) =0.2995, and Figini et al. (2017)=0.4093), the proposed algorithm scored a better Log Loss (proposed=0.2496).

Table 2: Summary of 1stand 2nd rounds of predictions

Algorithm		**1st Round**						**2nd Round**					
	After	**AU**	**F1**	**Pre**	**Rec**	**Mat**	**Log-**	**AU**	**F1**	**Pre**	**Rec**	**Mat**	**Log-**
Son et al (2019)	1st	0.77	0.73	0.73	0.73	0.40	0.52	0.87	0.82	0.82	0.82	0.64	0.39
	2nd Yr	0.78	0.73	0.73	0.73	0.41	0.50	0.88	0.83	0.83	0.83	0.65	0.36
	3rdYr	0.81	0.77	0.77	0.77	0.48	0.47	0.89	0.85	0.85	0.86	0.68	0.34
	4thYr	0.84	0.79	0.79	0.79	0.52	0.44	0.90	0.85	0.85	0.85	0.70	0.31
	5thYr	0.86	0.80	0.80	0.80	0.57	0.40	0.92	0.87	0.87	0.87	0.73	0.28
Figini et al (2017)	1styr	0.70	0.66	0.69	0.65	0.29	0.60	0.77	0.72	0.74	0.72	0.37	0.52
	2ndYr	0.73	0.67	0.70	0.66	0.30	0.59	0.80	0.74	0.75	0.73	0.41	0.49
	3rdYr	0.75	0.68	0.71	0.68	0.36	0.56	0.82	0.75	0.77	0.74	0.47	0.47
	4thYr	0.78	0.70	0.73	0.69	0.42	0.54	0.85	0.76	0.78	0.75	0.50	0.43
	5thYr	0.80	0.73	0.75	0.73	0.45	0.51	0.87	0.79	0.80	0.78	0.56	0.40
Chou et al (2017)	1styr	0.71	0.69	0.70	0.68	0.31	0.60	0.84	0.79	0.79	0.79	0.58	0.43
	2ndYr	0.72	0.70	0.71	0.70	0.33	0.61	0.87	0.80	0.80	0.80	0.62	0.39
	3rdYr	0.78	0.74	0.74	0.73	0.44	0.53	0.89	0.83	0.83	0.83	0.66	0.34
	4thYr	0.80	0.74	0.75	0.74	0.46	0.51	0.90	0.84	0.84	0.84	0.68	0.33
	5thYr	0.83	0.76	0.77	0.76	0.49	0.49	0.92	0.85	0.85	0.85	0.75	0.29
Proposed	1styr	0.76	0.74	0.74	0.74	0.40	0.53	0.87	0.83	0.83	0.83	0.65	0.36
	2ndYr	0.78	0.74	0.74	0.74	0.40	0.50	0.89	0.84	0.84	0.84	0.66	0.34
	3rdYr	0.81	0.76	0.76	0.76	0.47	0.47	0.91	0.86	0.86	0.86	0.70	0.29
	4thYr	0.84	0.79	0.79	0.78	0.52	0.43	0.92	0.88	0.88	0.88	0.74	0.26
	5thYr	0.87	0.81	0.81	0.81	0.57	0.39	0.93	0.88	0.88	0.88	0.78	0.24

As shown in Table 3, Figini et al. (2017) was the poorest classifier. Because of over-fitting, Chou et al. (2017) struggled, and as of Son et al. (2019), its Precision is not that high.

The usual better behaviour of the ensemble models is demonstrated. Data are highly unbalanced in our problem data. Besides, resampling of the minority class (Bankruptcy Companies) is a formidable solution to be executed. This is because certain features give a high level of sensitivity. Above all, the majority class, i.e., the companies currently active, holds a subset of them; however, they will become bankrupt in the next 5 years (usually, between 5% and10% once in every decade).

As demonstrated in Table 3, performing a second round boosted the AUC metric by at least 7%; for example, the AUC for the Son et al. (2019) model was 78% in the 1st round and grew to 88% in the 2nd round. Additionally, the enhancement of F1, recall, and precision confirms the same outcomes. Indeed, we can observe an increase in performance across all algorithms and time periods. The performance of the proposed is better than Son et al. (2019) among the EMs. The AUC score of our solution scored better results than those achieved if we consider the excellent results of our two research works, viz., proposed and Son et al. (2019). This is because only 15 financial and operational characteristics are used as individual variables, whereas more than 40 characteristics are used by other research articles. The characteristic selection and the vitality of financial data are anticipated by demonstrating the efficacy of our procedure.

Furthermore, a consistent anticipation rate of up to 5 years is presented compared with the other papers, where a maximum of 18 months was considered.

5.2. Classification: Direction of Nominal Returns

The nominal returns' direction is anticipated by assessing the classifiers. Table 3 contains the detailing of the related outcomes. While using no predictor, the benchmark's performance is reflected by the first row, that is, voting the majority class. Because of acute class imbalance, it leads to accuracy above the average level. The reason for making an analogy with balanced accuracy is also explained in the following analysis. Compared with the NB benchmark, the proposed method of Son et al. (2019) shows the progress of 4.7% points, which testifies the most excellent stable precision on the test set among the baseline models from conventional ML. The stemming from Figini et al. (2017) with 500 trees has resulted where 3 variables are sampled at each split. The strength of the model is highlighted as an *out-of-the-box* classifier.

Table 3:The classification for nominal returns

Method	Training Dataset	Test Dataset			Increase in Performance in the Test Dataset Compared to the Baseline		
	Accuracy	Accuracy	Balanced Accuracy	AUC	Accuracy	Balanced Accuracy	AUC
Son et al. (2019)	0.559	0.552	0.542	0.559	0.024	0.042	0.059
Figini et al (2017)	0.537	0.539	0.538	0.561	0.011	0.038	0.061
chou et al (2017)	0.541	0.550	0.526	0.557	0.022	0.026	0.057
Proposed	0.593	0.579	0.566	0.551	0.051	0.066	0.051

The conventional ML is outperformed by DL. For example, the proposed model's enhancement of 6.8% points is yielded above the other models [29], [30], [31], [32]. A meager amount of 0.1% points has increased due to word embeddings, but the AUC score increases by 0.5%. Consequently, the proposed model's performance is excellent among all techniques, which amounts to an overall improvement of 7.1% points. Otherwise, an addition of 0.8% points has resulted in the enhancement of the balanced accuracy (Fig. 8).

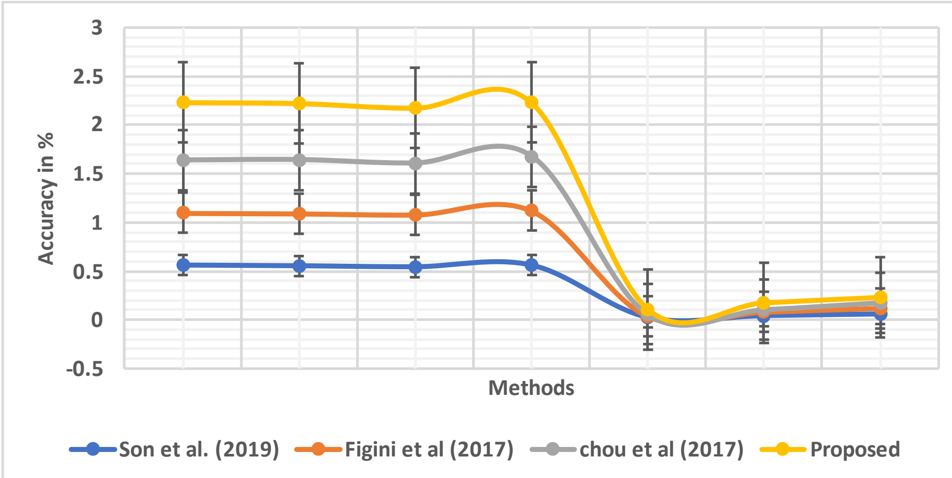

Fig. 8: The classification for nominal returns

5.3. Direction of Abnormal Returns

The direction of abnormal returns is anticipated using the results reported by Table 4, showing an overview, the same as the categorization of nominal returns [33]. The performance of Son et al. (2019) is better with a balanced accuracy of 0.542, which was outperformed by Figini et al. (2017) at a rate of 0.545. The latter attained a consistent precision of 4.5%, points more significant than the Chou et al. (2017) benchmark.

Table 4: Categorization of abnormal Returns

Method	Training Dataset	Test Dataset			Increase in Performance in the Test Dataset Compared to the Baseline		
	Accuracy	Accuracy	Balanced Accuracy	AUC	Accuracy	Balanced Accuracy	AUC
Son et al. (2019)	0.557	0.562	0.547	0.552	0.022	0.047	0.052
Figini et al (2017)	0.555	0.552	0.538	0.555	0.012	0.038	0.055
Chou et al (2017)	0.553	0.554	0.532	0.556	0.014	0.032	0.056
Proposed	0.576	0.578	0.564	0.577	0.038	0.064	0.077

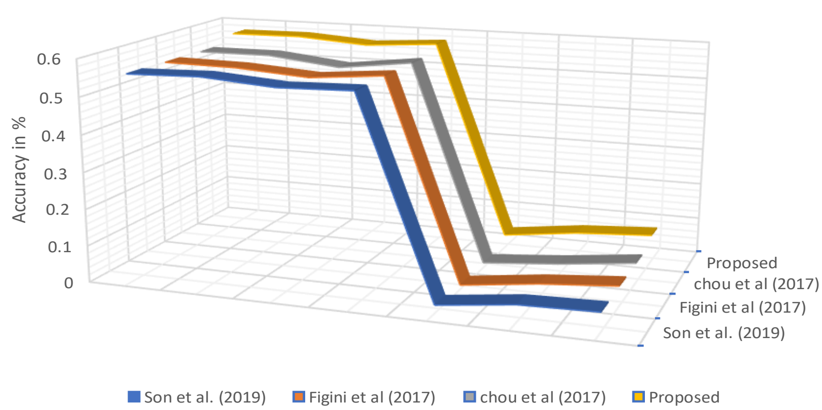

Fig. 9: Categorizations of abnormal returns

As far as DL is concerned, the performance above conventional ML models has failed to progress better. But in this task, the proposed model triumphs by a crossover of 5.6 points in the benchmark's balanced accuracy. The use of word embeddings has further improved the performance indicating the progress of 6.6% compared to the naïve model.

As GRU-CNN's show outstanding performance in terms of balanced accuracies, they show significant enhancements by 0.7% and 1.7%, respectively (Fig. 9). Further, by combining both pre-processing and word embeddings with GRU, the value gets an additional enhancement of 8.3%.

6.0 CONCLUSION

This research paper presents a prediction model for identifying default and non-default corporate customers in the banking business that employs DNNs (CNN + GRU) and TF-IDF pre-classifiers. The model aids bankers in deciding whether or not to allocate cash to a particular non-default client in a FI. The algorithm makes a highly accurate forecast about the client based on the credit score information. This study's major contribution is assessing and categorizing credit scoring techniques based on financial disclosure and other statistical data representing a customer's financial behaviour. The predictive algorithm would be highly beneficial to FIs in forecasting credit scores for current clients, saving more money for the lenders. The activation function of a DNN aids the predictive model in making a forecast, while the parameter tuning function improves accuracy. Additional data from other risk sources, such as cybersecurity and seismic data, will be introduced in the future, and the ML module will explicitly integrate the dynamic evolution of the system.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Science and Engineering Research Board (SERB), Department of Science & Technology, India, for financial support through the Mathematical Research Impact Centric Support (MATRICS) scheme (MTR/2019/000542). The authors also acknowledge SASTRA Deemed University, Thanjavur, for extending infrastructural support to carry out this research work.

REFERENCES

[1] D. Jung et al., "Conceptual Framework of an Intelligent Decision Support System for Smart City Disaster Management", *Applied Sciences*, Vol. 10, No. 2, 2020, pp. 666. https://doi.org/10.3390/app10020666

[2] L. Kee-hung et al., "Just-in-Time Logistics", *Gower Publishing Company*, ISBN:978-0-566-08900-8, 2009.

[3] M. Rashidi, et al., "Decision Support Systems", *Management of Information Systems*, 2018, https://dx.doi.org/10.5772/intechopen.79390.

[4] F. Erik, et al., "Fintech and the Digital Transformation of Financial Services: Implications for Market Structure and Public Policy". *The Bank for International Settlements and the World Bank Group*, 2021, pp. 117.

[5] Fed Survey, "Small Business Credit Survey–Report on Startup Firms". https://www.newyorkfed.org/medialibrary/media/smallbusiness/2016/SBCS-Report-StartupFirms-2016.pdf, 2017.

[6] E. Altman, et al., "Assessing the Credit Worthiness of Italian SMEs and Mini-bond Issuers". *Global Finance Journal*, Vol. 43, No. 10045043, 2020, pp. 1-21, https://doi.org/10.1016/j.gfj.2018.09.003.

[7] E. Altman, et al., "Financial and Nonfinancial Variables as Long-Horizon Predictors of Bankruptcy". *Journal of Credit Risk*, Vol. 12, No. 4, 2016, pp. 49–78.

[8] E. I. Altman, "Financial Ratio: Discriminant Analysis and the Prediction of Corporate Bankruptcy", *The Journal of Finance*, Vol. 23, No. 4, 1968, pp. 589–609.

[9] *E. I. Altman, "*Predicting Financial Distress of Companies: Revisiting the Z-Score and ZETA Models". *In: Bell, A.R., Brooks, C., Prokopczuk, M. (Eds.), Handbook of Research Methods and Applications in Empirical Finance*. Edward Elgar Pub, 2014, pp. 428–456.

[10] J. A. Ohlson, "Financial Ratios and the Probabilistic Prediction of Bankruptcy". *The Journal of Accounting Research*, Vol. 18, 1980, pp. 109–131.

[11] M. D. Odom et al., "A Neural Network Model for Bankruptcy Prediction". *IJCNN International Joint Conference on Neural Networks. IEEE*, 1990, pp.163–168.

[12] P. K. Coatset al., "Recognizing Financial Distress Patterns Using a Neural Network Tool". *Financial Management*, Vol. 22, No. 3, 1993, pp.142–155.

[13] J. E. Boritz, et al., "Predicting Corporate Failure using a Neural Network Approach". *Intelligent Systems in Accounting, Finance and Management*, Vol. 4, 1995, pp. 95–111.

[14] I. Brown et al., "An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets". *Expert Systems with Applications*, Vol. 39, No. 3, 2012, pp. 3446–3453.

[15] M. J. Kim, et al., "Geometric Mean-based Boosting Algorithm with Over-Sampling to Resolve Data Imbalance Problem for Bankruptcy Prediction". *Expert Systems with Applications*, Vol. 42, No. 3, 2015, pp. 1074–1082.

[16] S. Milusheva, et al., "Applying Machine Learning and Geolocation Techniques to Social Media Data (Twitter) to Develop a Resource for Urban Planning". *PLoS ONE*, Vol. 16, No. e0244317, 2021.

[17] M. Imran, et al., "Engineering Crowdsourced Stream Processing Systems". arXiv:1310.5463, 2013.

[18] M. Imran, et al., "AIDR: Artificial Intelligence for Disaster Response". Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion), Seoul, Korea, 2014, April 7–11.

[19] M. Bakerand et al., "Investor Sentiment in the Stock Market". *Journal of Economic Perspectives*, Vol. 21, No. 2, 2007, pp. 129-152.

[20] M. Obthong, et al., "A Survey on Machine Learning for Stock Price Prediction: Algorithms and Techniques". *Soton*, Vol. 1, No. 1, 2020, pp.1–12.

[21] O. Icanand T. B. Celik "Stock Market Prediction Performance of Neural Networks: A Literature Review". *International Journal of Economics and Finance*, Vol. 9, No. 11, 2017. pp. 100-108.

[22] S. Baniket al., "Hybrid Machine Learning Technique for Forecasting Dhaka Stock Market Timing Decisions". *Computational Intelligence and Neuroscience*, 2014.

[23] S. Banik et al., "Hybrid Machine Learning Technique For Forecasting Dhaka Stock Market Timing Decisions", *Computational Intelligence and Neuroscience*, Vol. 2014, Article ID 318524, 2014, pp.1-6, https://doi.org/10.1155/2014/318524.

[24] D. Rumelhart et al., "Learning Representations by Back-Propagating Errors", *Nature*, Vol. 323, 1986, pp. 533–536, https://doi.org/10.1038/323533a0

[25] Y. Bengio et al., "Learning long-term dependencies with gradient descent is difficult", *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, pp. 157-166. https://doi.org/10.1109/72.279181.

[26] S. Hochreiterand et al., "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[27] Son et al., "Data Analytic Approach for Bankruptcy Prediction". *Expert Systems with Applications*, Vol. 138, Article 112816, 2019, https://doi.org/10.1016/j.eswa.2019.07.033.

[28] S. Figini et al. "Solvency Prediction for Small and Medium Enterprises in Banking." Decision Support System, Vol. 102, 2017, pp. 91-97, https://doi.org/10.1016/j.dss.2017.08.001.

[29] C. H. Chouet al., "Hybrid Genetic Algorithm and Fuzzy Clustering for Bankruptcy Prediction". *Applied Soft Computing*, Vol. 56, 2017, pp. 298–316.

[30] K. Habeebah et al., "Diagnosis of Metabolic Syndrome Using Machine Learning, Statistical and Risk Quantification Techniques: A Systematic Literature Review". *Malaysian Journal of Computer Science,* Vol. 34, No. 3, 2021, pp. 221-241.

[31] K. Sathish Kumar et al., "Area-Based Efficient And Flexible Demand Side Management To Reduce Power And Energy Using Evolutionary Algorithms", *Malaysian Journal of Computer Science*, No. 1, 2020, pp., 61-77, https://doi.org/10.22452/mjcs.sp2020no1.5.

[32] M. Mandana et al., "An Extension of the Outlier Map for Visualizing the Classification Results of the Multi-Class Support Vector Machine". *Malaysian Journal of Computer Science*, Vol. 34, No. 3, 2021, pp. 308-323, https://doi.org/10.22452/mjcs.vol34no3.5.

[33] G. S. Bagale et al., "Small and Medium-Sized Enterprises' Contribution in Digital Technology", Annals of Operations Research, 2021, https://doi.org/10.1007/s10479-021-04235-5.