B-LSTM-NB BASED COMPOSITE SEQUENCE LEARNING MODEL FOR DETECTING FRAUDULENT FINANCIAL ACTIVITIES

Arodh Lal Karn[1], Karamath Ateeq[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]School of Information and Technology-BSIT, Skyline University College-1797, Sharjah, United Arab Emirates (UAE)

[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], karamath.ateeq@skylineuniversity.ac.ae[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.sp2022no1.3

ABSTRACT

Deep Learning (DL) in finance is widely regarded as one of the pillars of financial services sectors since it performs crucial functions such as transaction processing and computation, risk assessment, and even behavior prediction. As a subset of data science, DL can learn and develop from their experience, which does not require constant human interference and programming, implying that the technology will improve quickly. By loading an Ensemble Model (EM), a Deep Sequential Learning (DSL)model, and additional upper-layer EM classifier in the correct order, a new "Contained-In-Between (C-I-B)" composite structured DSL model is recommended in this article. In cases like Fraud Detection System (FDS), where the data flow comprises vectors with complex interconnected characteristics, DL models with this structure have proven to be highly efficient. Finally, by utilizing optimized transaction eigenvectors, a NB classifier is trained. This strategy is more effective than most standard approaches in identifying transaction fraud. The proposed model is evaluated for its accuracy, Recall and F-score, and the results show that the model has better performance against its counterparts.

Keywords: ***Sequential Learning, Fraud Detection, Deep Learning, Ensemble Model, Financial Institutions, FinTech***

1.0. INTRODUCTION

Across the world, around 5% of the income of Financial Institutions (FI) is lost to scam, whereas the money lost directly for scam vary in dollars ($). Loss in productivity and customer support (and probable abrasion) caused a hike in the actual cost, and the losses due to hidden scams are not to be indicated. Most banks use the systems for monitoring transactions for Fraud Detection (FD) activity. These systems are primarily domestic, and their software needs manual support. Yet, the traditional FD systems handle the detection of personal real-time and Point-of-Sale (PoS) fraud. However, in the scam

pie, that is just one slice. A study of FDM can associate a customer's behaviour throughout all contact avenues and products for identifying blasting situations, cross-channel scams, and social networks.

The future scam prevention proposes to provide caller authentication for customers through call centers, managing fraud by institutional FD using case management system throughout customer channels and accounts. The ongoing and completed transactions are protected efficiently from scams. Depending on the amalgamation of features within the data, all the records driven by metadata are deeply connected [1]. Later, standard objects are recognized and distorted using statistical methods for producing personal views of objects within networks. There is a generation of distinct and restricted networks within the data by characterizing activities and statistically appropriate associations.

New trend banking Fraud Detection Management (FDM) crosses beyond the usual customer perspective and enhances a comprehensive knowledge of customer behavior to furnish a wholesome idea of fraudulent activity that includes the concerned offenders and irrelevant channels. Each transaction follows a set of guidelines and analytical models, viz., opening an account, Automated Teller Machine (ATM)operations, Net banking, customer care inquiries, etc. As against the customer's suspicious behavior, the transaction activity contrary to the enterprise's comprehensive intelligence and the system's transaction activity is checked in real-time. The fraudulent activity within or throughout the channels is predicted exactly for most transactions using a score delivered by the system within milliseconds (*ms)*. The monitoring of these transactions doesn't inundate the practical decision-making and authorization despite the system's operation on billions of records. The detection and investigation of the current scam and prevention of the new scam are performed by running many customer accounts by FDM everyday end [2]. The account holders' recent update and their essential links are created and analyzed by the system.

The cost of a scam incurred by an organization is more than $1.5M, according to the report from the Association of Certified Fraud Examiners (ACFE). However, a security attack's complete influence on a business is just a part of the direct and indirect financial problem [3]. The Nature of scams reaches its extreme level, like loss of revenue and obstructing the customer's experience, tampers the company's goodwill, and causes functional letdowns. The risk of scams boomed in all industries at the beginning of the pandemic. According to the ACFE study, 79% of respondents reported an expand in the overall level of fraud (up from 77% in August and 68% in May), and 38% reported a considerable rise. This tendency is expected to continue in 2021.

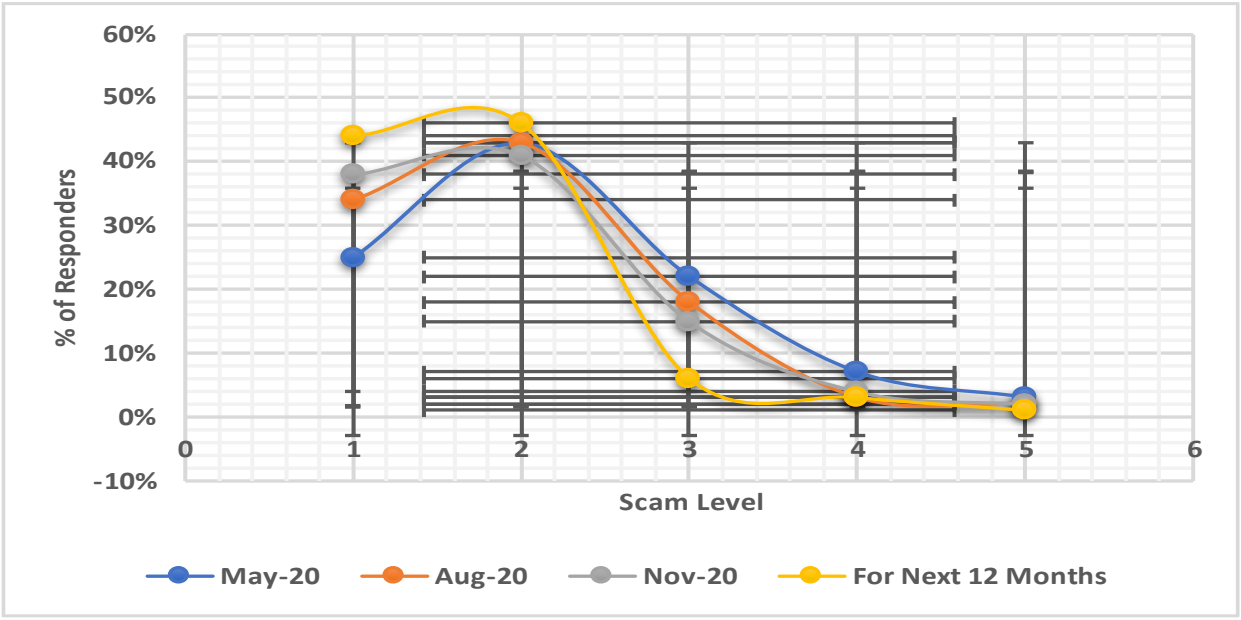

Fig. 1: Fraudulent activities Statistics

Bank employees usage of personal devices like smartphones, laptops, or tablets to access company information is one of the causes of unexpected inflation of scams. The Nature of cybersecurity is unidentical in corporate and home

environments as the Wi-Fi networks in the latter are easily prone to attack. Moreover, there will not be any periodical updating of an antivirus or anti-malware scan by bank employees, usage of easily detectable passwords, failing to get connected to Virtual Private Network (VPN), usage of online tools unapproved by the company, and much more [4].

The rest of the reasons for the upsurge of scam is shifting businesses to the public cloud and maximum usage of IoT devices without paying proper attention to security. In recent years, firms across all industries have seen an increase in the number of fraud events, which can be attributed to big online data transactions, lax security measures, and ineffective fraud protection systems [5]. The nature of fraud has altered significantly during the past year.

Targeted victims have taken a paradigm shift from single entities and small-scale businesses to large-scale ventures. The deformities caused by the coronavirus were misused by the attackers and unearthed new security threats that many companies are susceptible to.

In the new reality, traditional fraud detection approaches are no longer practical. The approaches to fraud risk management should develop as criminals' attacks get more sophisticated. It is important to learn about the different sorts of fraud events that might damage the business and how can the big data market analytics and Machine Learning (ML) be used to improve fraud detection. The benefits of using market analysis of big data and ML to detect fraud have already been realized by many firms across industries. By integrating ML, the following benefits are gained: (i) Cost-efficient and increasing profit, (ii) Giving better customer service, (iii) Identifying scam before it takes place, (iv) Expediting the investigation of scam by preventing manual work,(v) Enabling the identification of the system's flaws/business transactions [6].

The cases of banking sectors or a Financial Technology (FinTech) company that fall prey and incur heavy losses are thousands. The collection of money from individuals and firms happens in many ways:

- Insurance claim scam;
- Loan scam;
- Stealing identity;
- Stealing account details;
- Money laundering
- Credit card scam
- Smartphone scam
- Internal and External scam

There are a few idiosyncrasies in each sort of FinTech scam; however, they follow a similar pattern of algorithms, tools, and methodologies to encounter them. But still, *back-to-back* solutions are not available. Text-based solutions capable of processing and assessing chunks of text in corporate catalogues, commenting on the posts, etc., are needed in some instances like identity and account trafficking. ML-based technologies such as Optical Character Recognition (OCR) for recognizing images solve huge issues related to image-based FD [7].

In short, data about people, their behavioural patterns, variances, and much more are received or generated by an ML-based system. Then, under the category of scam, non-scam/a possible scam, the data are classified by the system using algorithms, and in a prescribed form, the result is presented as a report for managers, order to a third-party system, and much more. In alliance with several platforms and algorithms, an inclusive building process for the transaction of the Fraud Detection Model (FDM) has been proposed in this paper. Integrating ensemble and DL methods has presented an inventive "Contained-In-Between" (C-I-B) composite-structured sequence learning architecture. Especially in transaction FD where the vectors make the sequence with intricate correlated characteristics, electrifying performances have been displayed by models of identical structures.

Further, the article is systematized as follows: Section 2 presents the related research study, and followed by it is Section 3 that describes the prelims needed for understanding. Section 4 presents the recommended IB-LSTM-NB methodology, and Section 5 concludes the work.

2.0 RELATED WORKS

With the help of NNs, for detecting the fraudulent transactions, multiple approaches have been provided. Ghosh and Reilly proposed a Neural Networks (NNs) based FDM for Credit Card (CC) [8]. For FD, their proposed system was fitted in a bank. Based on NNs, Brause et al. [9] proposed a prediction model for CC card scams. A technique known as CARD WATCH was submitted by Aleskerov et al. [10] for a similar purpose, and it offers a Graphical User Interface (GUI) for distinct commercial databases. The NN model is the nitty-gritty of this system, and it furnished resounding FD rates.

A FDM based on NNs was proposed by a further study [11], and the NN is based on confidence. There was a proposal for a guaranteed, effective and accurate technology for FD in this model. If the transaction had low confidence, then it is deemed to be fraudulent. Syeda et al. [12] developed an identical Granular Neural Network (GNN) to detect scams related to a CC. Based on a collaborative and a sliding-window approach, two FDMs were designed by Dal Pozzolo et al. [13] for a concept-drift modification. The same ensemble of authors took conceptual drift into account in a new task. Ma et al. [14] also applied the same concept where for NNs' update, the authors proposed an incremental virtual learning method. In order to manipulate concept drift and data disproportion, an ensemble of identical and incremental learning has been offered by [15].

Anomaly detection in Fintech is the subject of multiple survey articles that provide a good overview of current trends. Ngai et al. [16] reported the earliest comprehensive inspection of intelligent systems for financial fraud detection. The survey conducted by Ahmed et al. [17] provides an overview of anomaly detection approaches in the financial domain, specifically clustering algorithms. Besides, a review of the methods for detecting anomalies in big data is provided in financial markets. Consecutively, by implementing partition and hierarchical-based clustering algorithms, presumptions on detecting anomalies and works are briefed by[18]. On FDM, Abdallah *et al.* proposed a survey [19]. Besides, in general, a complete survey on FinTech was presented by Gai et al. [20]. A survey on CC- FD was introduced by Ryman-Tubb et al. [21]. Subsequently, survey results on applying a classification-based approach to financial FD were proposed by West and Bhattacharya [22]. Besides, the merits and demerits of the classification-based system to financial FD of the present works like applied algorithms, performance, and kinds of scams are analyzed. Pourhabib *et al.* [23] proposed a general idea about Graph-based methods for detecting anomalies. In the FinTech domain, one of the newly studied techniques is known as Long Short-Term Memory (LSTM).

When comparing to usual transaction sample of an account, specific algorithms that are based on behaviour like Hidden Markov Model (HMM) and Peer Group Analysis (PGA) have been suggested for FD by detecting abnormalities [24], [25]. But for each account, there should be a construction of most behaviour-based models. The account of the usual historical patterns that are hard to acquire is relied on by these models. Recently, in sequence analysis work, Recurrent Neural Networks (RNN) based DL methods have been validated to perform well [26]. RNN uses sequence analysis techniques to analyze dynamic temporal behaviours for multiple accounts [27]. There is the inadequacy of feature learning capability within a one-time transaction for RNN models, though the extraction of sequential information between transactions is like most sequence analysis methods. Specific classification models such as Random Forest (RF) learn these associations within a single transaction very well by reducing SL.

For transacting FDM in association with several platforms and algorithms, a holistic building process has been proposed by [28]. The ensemble and DL methods are combined in order to present an inventive "Within→Between→Within" (WBW) sandwich-structured sequence learning model. Especially in situations like transaction FD, outstanding performances will be displayed by models in same structures, where vectors with intricate correlated attributes make the sequence. Olowookere et al. [29] conducted another study where cost-sensitive and ensemble learning paradigms that enhance FD in unnecessary datasets are contained in the proposed model. From the Decision Tree (DT), Multi-Layer Perceptron (MLP), and K-Nearest Neighbours (K-NN) algorithm, a cost-sensitive ensemble is constructed. Raghavan & Gayar, 2019 [30] conducted related studies, and FD is compared with autoencoder, K-NN, Support-Vector Machines (SVM), K-Means, Naive Bayes (NB), and RF. The authors concluded a study that SVM is ideal for large datasets, and for getting an outstanding result, it can be merged with Convolutional Neural Network (CNN), and K-NN and RF provide improved outcomes for small training datasets. However, for detecting scams, this study is restricted to supervised learning.

For detecting scams and efficiently listing the merits and demerits of these models, Amarasinghe *et al.*,2018 [31] conducted comprehensive research that contained supervised ML-like RNN, Fuzzy Logic (FL), Bayesian Network, and

SVM, and unsupervised ML-like hidden Markov Model, K-means clustering, and point outliers. Compared to FL, an Artificial neural network (ANN) gives a better precision of 33%. Moreover, ANN with a Genetic Algorithm (GA) is proposed by this research paper so that another ML algorithm is compared with it. Various supervised ML algorithms such as RF, LR, KNN, NB, DT, SVM have been analyzed by [32] and compared these models' sensitivity, exactness, and precision to FD for which scoring rule is the base. With the help of feedback and delayed supervised samples, the models are proposed to be trained, and subsequently, each probability will be aggregated to FD.

Logistics Regression (LR) and RF have been investigated by the authors [33] in another research, and for a better outcome, the characteristics and subsample ratios for unnecessary datasets are also assessed. The detection of anomaly to FD has been executed in the paper by [34]. Though the algorithms' precision goes up to 99.6%, isolation forest and local outlier components are applied in the data. However, when the whole dataset is taken for training the model, the accuracy is just 33%. An imbalanced dataset is a cause for achieving greater accuracy, and more FD transactions cannot be detected if the precision is less. To handle an imbalanced dataset, a framework has been proposed by Blagus et al., 2013 [35] using SMOTE function. The usage of bootstrapping and K-NN makes the Synthetic Minority Over-sampling Technique (SMOTE) oversample the minority class data for generating minority class' further synthetic observations that are the scam data since the number is low.

3.0 PRELIMS

3.1 Bootstrap Aggregation (BA)

Bagging and boosting are the categories of ensemble ML. The technique of Bagging benefits the classification of both regression and statistics. D_t is incorporated with Bagging, where the models' consistency is raised significantly to enhance accuracy and reduce discrepancy that ignores the challenge of overfitting.

Multiple weak models are taken by bagging in ensemble ML, gathering the predictions for the ideal forecast. To achieve the principle objective, different segments of the feature space are specialized by weak models that allow bagging leverage predictions to obtain from each model. Aggregation and bootstrapping are the two main components of bagging. Bootstrapping is a sampling approach that uses the replacement method to select a sample from a more extensive set. The algorithm is then applied to the samples. Using bootstrapping, the selection process is made absolutely at random through the use of sampling and replacements. In the absence of replacement, successive selections of variables are non-random since they are influenced by the preceding selections. The final forecast is based on the aggregation of model predictions to take into account all conceivable outcomes. Aggregation can be performed on the total number of outcomes or on the probability of predictions produced from each model's bootstrapping (Fig. 2).

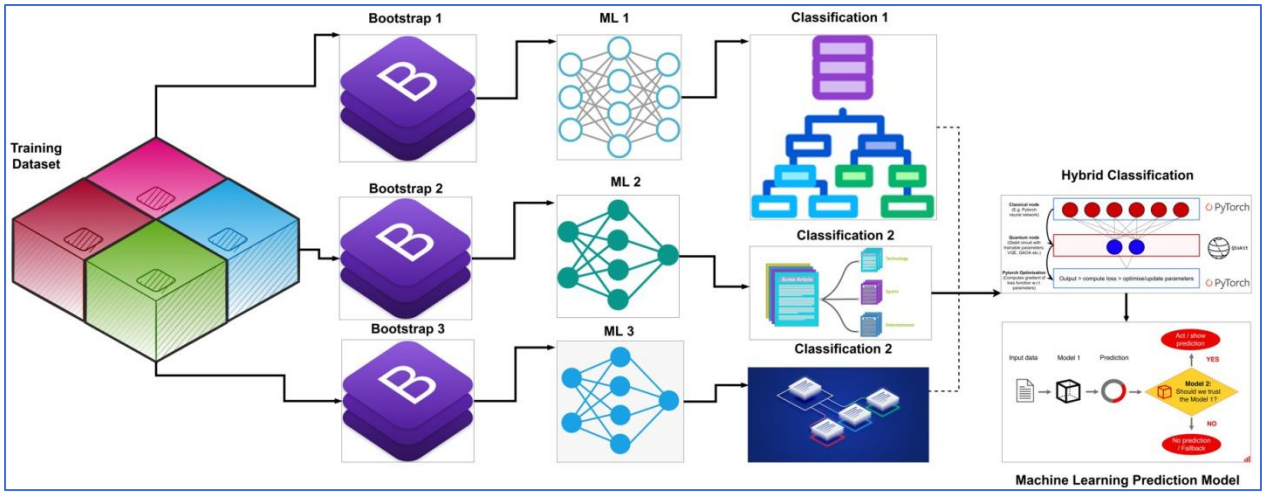

Fig. 2: Bagging (bootstrap aggregation) data flow diagram

The most significant EM is formed by both Bagging and Boosting. The ML algorithm is applied for training the multiple models with the help of ML, which is an EM. In a more significant ensemble of multi-classifiers, the ensemble method participates. The multi-classifiers are an ensemble of multiple learners that can run into thousands with a common aim to merge and solve a common problem. Hybrid methods are another type of multi-classifiers. Unlike the multi-classifiers, a set of learners is used by the hybrid methods though they can deploy different learning methods. The flaws caused mainly due to noise, bias, and variance, are the challenges faced by learning. ML's accuracy and consistency ensure ensemble techniques like bagging and boosting. In particular, when there are inconsistent classifiers, multiple classifiers' grouping minimizes variance compared to single classifiers. Multiple classifiers significantly present very reliable results. The base learner algorithm is selected first for which, the selection of either Bagging or boosting is required. For instance, boosting and bagging would be a group of trees and the size of which is equal to the user's preference if a classification tree is chosen.

The bootstrap approach is related to Bagging, whereby the training sets are selected randomly by getting replacement from the actual examples. As a result of resampling, there may be a possibility of various records for more than one time by following this procedure, whereas in the training set, others may not appear. Significantly, the non-concentration of training subsets by random selection with replacement is the demerit of typical bootstrap method. Thus, the ML algorithm cannot concentrate on the hard-to-classify instances to reduce the training errors because they may not exist in the training sets. Tüysüzoğlu et al. 2020 [36] proposed bootstrapping to overcome this problem since it encourages the correct classification of misclassified samples in training sets by ensuring its presence.

With the class output as y_1, assume the dataset $D = \{O_1, O_2, ..., O_n\} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and p – dimensional descriptive variable for the i^{th} object O_1 as x_i Hence, the number of objects (cases) in the dataset is '*n'*. Accept that $y_i \in Y = \{1, ..., k\}$, If there are *k* different class labels.

3.1.1 Definition 1 (Bootstrapping)

If n data instances are represented as, Eq. (1)

$$D = \{O_1, O_2, ..., O_n\} \tag{1}$$

Then, with the similar data size *'n'*, the sample generated in random and with the same percentage chance ($^1/_n$ for each comment), Eq. (2)

$$D^* = \{O_1^*, O_2^*, ..., O_n^*\} \tag{2}$$

is known as the bootstrap sample/resample, and it is represented by adding a star to the symbols. 0/more examples are denoted by the asterisk '***', indicating that more repeats are possible in '*D*'*. Likewise, '$\bar{o}$***' is the mean of the resample data, similar to'$\bar{o}$*'* is the mean of the real dataset. We obtain *b* individual bootstrap samples to form an ensemble *E* that is represented by Eq. (3)

$$E = \{D_1^*, D_2^*, ..., D_b^*\} \tag{3}$$

Every instance contains a probability (1-1/*n*)*n* for a given dataset of *n* examples that are not incorporated in the bootstrap sample. The probability approaches are 1/e = 0.368 if *n* is huge, and the non-selection of actual examples of 36.8% is known as *out-of-bag* instances, Eq. (4)

$$Lim_{n\to\infty}(1-\tfrac{1}{n}) = \tfrac{1}{e} \approx 0.368 \tag{4}$$

Also, it indicates that non-participation of the "tough" instances is about 36.8%, and because of this, a conciliation between bias and variance on classification may not be offered by the basic learning algorithm. A new bootstrapping method is known as e-Bootstrapping that prioritizes the challenging examples to overcome this limitation.

3.1.2 Definition 2(e-Bootstrapping)

With size *n*, the given dataset is as follows: Eq. (5)

$$D = \{O_1, O_2, ..., O_n\} \tag{5}$$

In the pre-training step, a *prior classifier* identifies the misclassified cases and are embodied by *M*, Eq. (6)

$$M = \{O_1, O_2, ..., O_m\} \tag{6}$$

The number of inaccurately classified cases is '*m'*, which is lesser than case size '*n'*. The remaining dataset is the accurately classified cases, represented by *C*, Eq. (7)

$$C = \{O_{m+1}, O_{m+2}, ..., O_n\} \tag{7}$$

where, $D = M \cup C, and, M \cap C = \phi$

A prediction error-based bootstrapping method is known as e-Bootstrapping. The entire misclassified cases in *M* and certain accurately classified examples extracted with additional from *C* are included in a dataset generated by e-Bootstrapping.

3.2 Improved Bagging (I-Bagging)

The replacement of the bootstrap method by I-Bootstrap enables I-Bagging to enhance the conventional bagging technique. The generation of training sets with the prospective for selecting hard-to-classify instances that the prior learner misclassifies is a major difference. There are four steps in the proposed I-Bagging technique, as shown in Figure 2.

- **Step I-Training Dataset:** On the real dataset, pre-training, a prior classifier is employed, and further, the dataset is classified as the accurately classified instances as one part, and the inaccurately classified models as the other type.

- **Step II-e-Bootstrapping:** By moving the misclassified instances and resampling them by getting a replacement from classified examples, various training sets are constructed by e-Bootstrapping. Hence, in each data subset, there is always an inclusion of tough instances. Diversity and permission to the learning algorithm for concentrating on hard-to-classify examples are provided by Step 2, and thus, it offers us a sufficient starting point.

- **Step III – Training:** The training is given to the base classifiers on distinct subsets of the training patterns. This study plans to use pre-training and training steps to be the same in ML algorithms; however, various ML algorithms can also be executed in future research. In other words, they are the ML algorithms of prior and base classifiers.

- **Step IV- Data Aggregating:** By deploying predominant voting to the outputs of each EM subset, a final prediction is made, and the base classifiers perform the fundamental classifiers and the classification task.

The classification of the hard-to-classify examples is also attempted by boosting technique viz., AdaBoost algorithm, and the easy-to-classify is ignored. But there is a demarcation between the enhanced bagging and boosting method. First of all, there is a simultaneous generation of training sets by I-Bagging from the actual dataset, and therefore, like boosting, it is not an iterative approach. Secondly, like expanding, each value is not assigned any weight values by I-Bagging; instead, all tough instances are directly copied into all training sets.

3.2.1 Algorithm for I-Bagging

It is an Enhanced Bagging (e-Bagging): A Novel Approach for EM

Step 1. Inputs

Training Dataset $of\ D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ where $y_i \in Y = \{1, \dots, k\}$
L: Prior and base classifiers' learning algorithms
t: Factors of the EM
n: Count of Scenarios
k: Class Size
x : A Classifiable Unlabeled data Evidence

Step 2. Method: **I-Bagging**

Step 3. Pre-Training: **Dataset**

Step 4. "D" to construct prior classifier using learning algorithm "L"

H ← L_{Train} (D);

Step 5. Create New datasets

C ← Ø, M ← Ø; i ← 0;

Step 6. Do

```
{
        Classify Instances Accurately
        If (h(xi) ==yi) Then
        {
                C.Update(xi,yi);
        }
        Else
        {
                M.Update(xi,yi);
                i ← i+1;
        }
} While (i < n);
```

Step 7. e-Bootstrapping

Step 8. Do

```
{
        Di ← Ø;
        Di ← Di ∪ M;
        Do
        {
                r← Random.Next(I,C.Length);
                Di.Add(Ci);
        } While (D.Length ≠ m);
        i ← i+1;
}While (i<t);
```

Step 9. Training

$h_i = L(D_i)$;

Step 10. Combining

Arriving at the Final Hypothesis

H(x) = Voting($h_1(x)$, $h_z(x)$, …, $h_n(x)$;

$$\boldsymbol{argmax} \sum_{\boldsymbol{y} \in \boldsymbol{Y}}^{\boldsymbol{t}} \boldsymbol{i} : \boldsymbol{y} = \boldsymbol{h_i(x)^l}$$

Step 11. End Process

3.3 Recurrent Neural Network (RNN)

In 1980s, RNN was first introduced [37], [38], [39]. An input and output layers are contained in a model. With the notion behind utilizing chain-like networks of reiterating modules as a cloud memory for storing information from previous processing steps, RNNs have chain-like networks. The sequence of inputs is adopted by including a feedback loop by the RNNs that permit the NN, unlike Feedforward Neural Networks (FNN). This means that to cause an impact of $Step_1$ and the next step, output from $Step_{t-1}$ is sent into the network. Thus, in learning sequences, RNNs have been efficient.

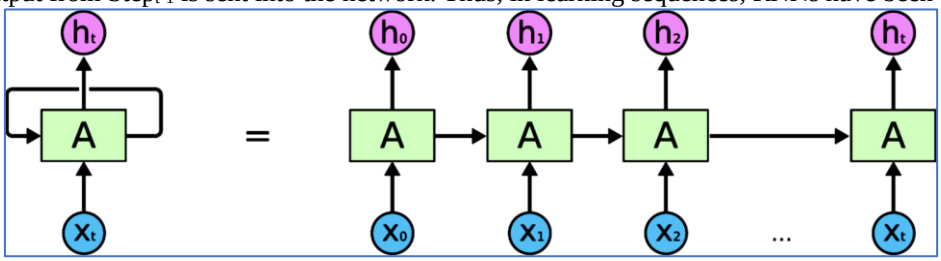

Fig. 3: RNN architecture

Fig. 3 shows a whole network built from a simple RNN with one input and output and recurrent hidden unit, where X_t represents the input at time step t and h_t represents the output at time step t. RNNs train using a backpropagation method, which is a widely used algorithm for computing gradients and modifying weight matrices in ANNs.

Nonetheless, followed by the change of the feedback process, the weights are adjusted, and so, it is denoted as the Backpropagation Through Time (BPTT). According to the computed portion of the units of the total output error, a layer-by-layer working-backwards method from the ultimate output of the network is used in the BPTT process to modify each unit's weights. During the updating process, the gathering of error gradients causes the recurrence of information loops leading to large updates to NN model weights and an inconsistent network. Hence, due to the gradient disappearing and the blasting gradient issues, the learning of a pattern from Long-Term Dependency (L-TD) cannot be performed efficiently by BPTT. The problems in the training of repeated NNs are caused by one of these significant reasons.

3.4 Long Short-Term Memory (LS-TM)

By including further communication per module (or cell), Hochreiter and Schmidhuber [40] introduced LSTM, an advancement of RNN for addressing limitations, as mentioned earlier. LSTMs are an exclusive type of RNN proficient in learning L-TD for recounting information for an extended time as a default. In the form of a chain model, the LSTM model is systematized according to Olah, 1999 [41]. But a distinct design is contained in the recurring module. This LSTM module has four interacting layers integrated with an innovative method of communication rather than a single NN as a typical RNN. Figure 4 shows the structure of the LSTM-based neural networks.

Memory blocks known as cells are contained in a conventional LSTM network. The cell and hidden are the two states being shifted to the subsequent cell. The primary chain of data flow is the cell state that permits the data flow to forefront, necessarily unaltered though some linear transformations may occur. Through sigmoid gates, the data can be attached or detached from the CS. A layer or a sequence of matrix functions is the same as a gate with various individual weights. Since LS-TMs use gates for controlling the memorizing process, they are designed to prevent the issue of L-TD. Identifying unwanted information is the preliminary step in building an LSTM network where there is an omission from the cell. The sigmoid function determines the detection and deletion of data in this process that extracts the final LSTM (h_{t-1}) unit's output at time *t-1* and new input (X_t) at time *t*. Moreover, the determination of the sigmoid function insists on which the previous value is eliminated. This is known as forget gate (f_t), where the vector f_t contains the limits of importance from '0' to '1' parallel to every count in CS, C_{t-1}, Eq. (8).

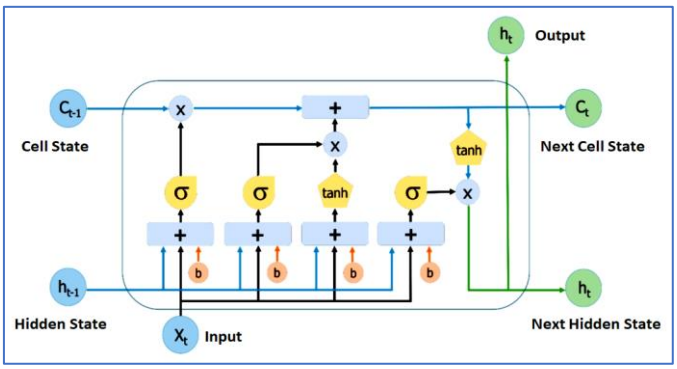

Fig. 4: The structure of LS-TM neural network

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (8)$$

The sigmoid function is σ, and the weight factors and bias are respectively W_f and b_f of the f_t. The determination and storing data from the new input (X_t) in the Cell State (CS)and the updating of the CS are performed in the following step.

Tanh is the $sinh^{-1}$tangent f(x)(Figure 5) and is the $sinh^{-1}$analogue of the Tan circular $f(x)$used within *'θ (theta).'* The ratio of the associated $sinh^{-1}[sin(\theta)+f(x)]$ and $sinh^{-1}$ $[cos(\theta)+f(x)]$is described as Tanh[α],α can also be presented as sigmoid, where Log is the natural '$log_b\, x$.

The sigmoid and tanh functions are the two layers contained in this step. Initially, the requirement for updating the new information or ignoring it is decided in the sigmoid layer (0/1), and secondly, the values are given weightage by the tanh function to determine their Nature of importance (-1/1). The new CS is updated by multiplying the two values. Later, the new and existing memory C_{t-1} are combined, thus resultant in C_t, Eq. (9), Eq. (10), and Eq. (11)

$$i_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (9)$$

$$N_t = \varnothing(W_n[h_{t-1}, X_t] + b_f) \quad (10)$$

$$c_t = C_{t-1} f_t + N_t i_t \quad (11)$$

Here, at time *t-1* and *t*, C_{t-1} and C_t are the CSs, whereas the weight factor scenarios and bias of the CS are W and b, respectively. Though the resultant values (h_t) depend on the consequent CS (Ot), it is filtered. First of all, the elements of the CS that become the output are determined by a sigmoid layer. Then the tanh layer that creates the new values from the CS (Ct) is multiplied with the result of the sigmoid gate (Ot) with value limits between -1 and 1, Eq. (12) and Eq. (13)

$$O_t = \sigma\{W_0(h_{t-1}, X_t) + b_0\} \quad (12)$$

$$h_t = O_t \varnothing(C_t) \quad (13)$$

Here, the weight matrices and bias of the output gate are W_o and b_o, respectively.

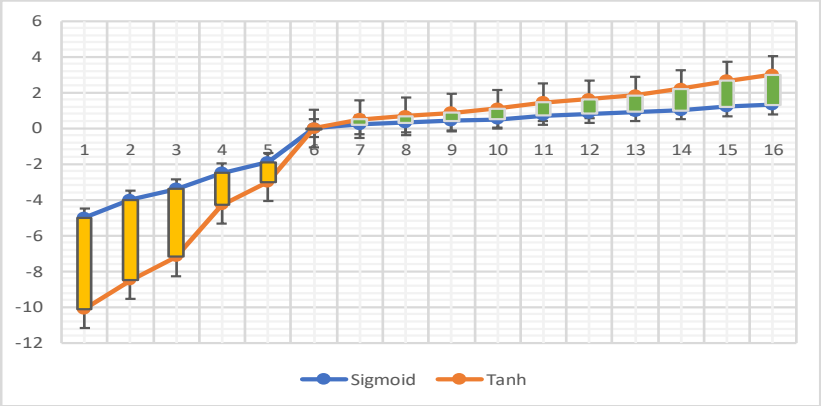

Fig. 5: The Hyperbolic Tangent

3.5 Naive Bayes (NB)

Using the Bayes' theorem, the probability that belongs to class-based is calculated by a statistical method known as Bayesian classifier. The reason to be called naïve is that the possibilities of independent features are separated, which is impossible to occur in the physical world. An already occurred event is assumed for calculating the possibility of an event to occur, Eq. (14)

$$P_r\left(\frac{c}{X}\right) = \frac{P_c\left(\frac{x}{c}\right).P_r(c)}{P_r(x)} \tag{14}$$

where,

P(c), P(X|c), and P(X). the computation of the target class cP(c|X)'s Posterior Probability (PP)is done Target class's PPc P(c|X) is premeditated from P(c), P(X|c), and P(X).

3.6. Algorithm for Naïve Bayes

The Naïve Bayes (NB) is an intuitive method that makes predictions using the conditional probabilities of every attribute belonging to each class. The Bayes' Theorem is a formula that counts the frequency and amalgamation of values in the historical data for calculating a probability. The maximum probability technique is used by parameter estimation for NB models. NB model frequently performs well in various complicated real-time applications despite over-simplified inferences. NB theorem's one of the main merits is the requirement of less quantity of training data sets for estimating the parameters.

Step 1. Input
Step 2. A group of tuples = D
Step 3. There is an n-dimensional attribute vector for each of the Tuples
Step 4. $X=(x_1,x_2,x_3,\ldots x_n)$
$C_1, C_2, C_3,\ldots C_m$ classes are all that are permitted
Step 5. According to the Naïve bayes classifier, X subclass C_i Iff.
Step 6. $P(Ci/X) > P(Cj/X)$ for $1 < j <= m, j \propto i$
Theory of Optimum Posteriori= $P(Ci/X) = P(X/Ci)P(Ci)/P(X)$
Step 7. Optimum $P(X/Ci)P(Ci)$ as $P(X)$ is constant
With numerous features, analysis is highly parallelized, $P(X/Ci)$
Naïve bayes classifier hypothesis of *"Class Conditional Independence"*
Step 8. $P(X/Ci) = \prod_{k=l}^{n} P(x_k/Ci)$
$P(X/Ci) = P(x1/Ci) * P(x2/Ci)^* \ldots^* P(xn/Ci)$
Step 8. End

4.0 PROPOSED METHODOLOGY

4.1 Contained-In-Between (C-I-B)-FDM

At present, the transaction scam risks are detected by the FI using two methods, viz., Rule-based and ML-based algorithms. The rule library is persistently established and renewed to make the rule-based method work that depends on the features of transaction behaviour. Moreover, the querying of the rule library will differentiate the actual risks in the transaction at the time of transaction. For example, a match will be conducted by the rule library to find out any anomaly features in this transaction behaviour when a huge payment is made at a convenience store. It is based on the extraction and standardization of the past experiences of transaction scams. The financial experts' knowledge is predominantly relied upon to comprise subjective components and certainly leads to omissions. Therefore, in this day, wherein in all insufficient shapes and forms, the transaction scam occurs.

The scam is handled more objectively, and precise methods are used by the ML-based process comparatively. Better learning of certain basic scam models is granted by certain kinds of good classification algorithms like LR, RF, and Gradient-Boosted Decision Trees (GBDT)and implemented in various FI. But feature engineering may be highly complicated in a real transaction situation. For example, if a lump sum amount is paid at a convenience store at midnight using a CC that has not been used for long or an expensive medicine is prescribed for a patient with viral flu, frequently visit a doctor. In those cases, there will be an occurrence of complicated characteristics like *"Long_Been_Unused","Big_Sum", "Viral_Flu", "Long_Been_Unused",* and *"Late_Night"* that is challenging to CNN methods.

4.1.2 Artificial Feature Engineering (AEF)

In authentic transaction fields, every transaction must be primarily plotted into a row vector. The vectors, as mentioned earlier, require more AEF. Skills derived from statistical methods like rolling-window and Recency-Frequency-Monetary (RFM) model helps in calculating many derivative variables, which are found through previous occurrence. For instance, the quantity of current and last deals and their variance acts as a current transaction's features. Distinct features are computed from total amount above separate time duration. Based on our analysis, few favorable characteristics are used as features. For example, through the historical dealings, if lots of fake cases occur in a particular location, it shows a modified feature *"Is_High_Risk_Loc"* as '1', or else it will show '0'.

This type of model allows random numerical variables such as money transactions. In addition to that, its efficiency is further improved through artificial analysis. To variate them with more differences, Weight of Evidence (WoE) is best. Features with perfect quality are attained through spark libraries such as ML/MLib. For example, the "*Vector_Indexer*" Application Programme Interface (API)automatically detects location and merchant type using numbers and makes an index. *"String_Indexer"* API is more advanced as it does the above task, and through frequency, it orders and commendably enhances the model's performance.

A novel and outstanding multi-layered hybrid FDM proposed by us intensify the feature correlation learning and transaction order. IB→LSTM→NB is the composite structure of the framework. Figure 6 shows the unitary transaction in a single RNN method, and its insufficient feature learning capacity is focused on primary importance, and using the LSTM model framework front end is optimized. LSTM network takes both optimized and artificial features as input, and in a single transaction, it enhances sequential and inter-sequential features. This structure cannot utilize the LSTM network outcome to differentiate FD, and it does not need an inter sequence feature learning level. To obtain a final transaction feature vector, both inter-sequence features are merged with the original features. Finally, to differentiate the fraud, the integrated learning features are attached to the uppermost surface of the NB model as a final classifier.

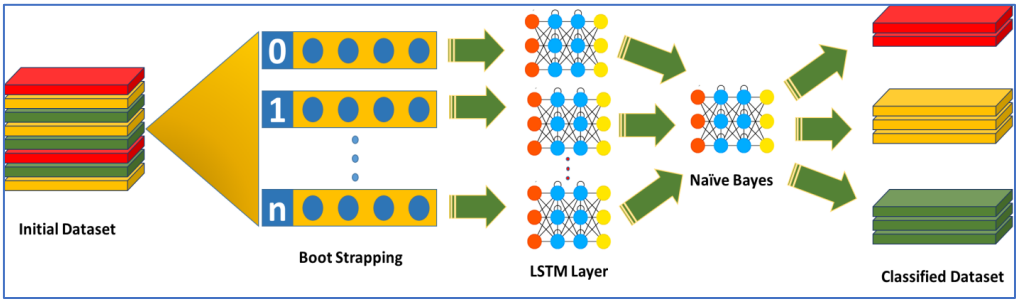

Fig. 6: IB→LSTM→NB architecture

The sandwich-structured FDM can study the linear correlation, the same as amid the integrated attributes 1 *(Late_Night+Small_Sum*) and integrated attributes 2 (*Late_Night+Big_Sum*). It is obtained by guessing that the account exhibited several minimum payments pursed by a large sum and the prior "IB→LSTM". Such a method is termed the "*Sequencing of Integrated Features (SOIF)*". In addition to that, the following "LSTM→NB" method can also study the unusual attributes, which is the same as the union of linear characteristics1 "*Testing_of_Various_Minimum_Sums+Large_Sum_Cash_Out*) and linear attributes 2 (*Present&Preceding_Trans_Regional_Transaction_Positions*). Such a process that "*SOIF*" is based on is the feature learning ability of the system.

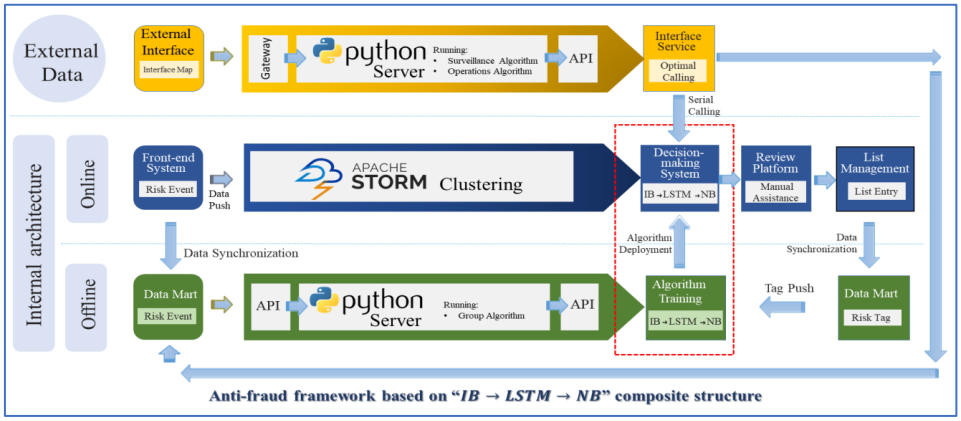

Fig.7: Complete SDM based on (C-I-B) DSL architecture

The last FDM may be trained by utilizing a classifier in the upper layer, which depends on the advanced eigenvectors Vop. The classifiers are selected from the standard algorithms for this process, but the recommended models are IB, RF, GBDT, and eXtreme Gradient Boosting (XGBoost). These are the most efficient models in FD domain. The boosting EMs may not be preferred to incorporate the benefits of multi-methods since IB has been included in the preceding feature learning process. Therefore, the IB model is chosen as the uppermost layer of the classifier, and it is the finest altogether method that may be executed later. Here, the entire system training process is analyzed. Initially, IB model is utilized to enlarge successful features in solitary transactions. Afterwards, to study the linear features, the LSTM representation is applied. Finally, the NB representation is implemented in the enhanced Eigenvectors number of provisional order (Vop) for every transaction to investigate many efficient components. To make complex structures, the linear learning method and the analogous infrastructures shall be combined. The benefit of C-I-B design is naturally explainable.

Other than fraudulently computed highlights, the highlights obtained from gathering models like IB may be consecutively subordinated between exchanges. For instance, following a few uncertain *"small sum off-site swaps at the middle of the night*," "*a massive amount of off-site transaction at midnight*" would occur. The primary C-I-B formation can naturally be studied at similar doubtful extortion strategies at other levels with linear dependencies. In the meantime, recently studied linear features can have effective relations with others within the conversation. For example, a "*present huge amount of*

transaction occurs later than few faltering ones" pattern could be more untrustworthy if merged in another pattern similar to "*present transaction location is varied from the preceding uncertain ones*". The linear features shall be reunited to make novel features in a distinct transaction. To study the comparable data, the second C-I-B structure is implemented. A complete design of the characteristic processing flow is exhibited in Figure 7.

5.0 EXPERIMENTAL RESULTS

5.1 Training Dataset

Paysim engineered dataset provides this dataset of portable money exchanges distributed on Kaggle. This data comes from the Paysim synthetic dataset of mobile money transactions, which was recently published on Kaggle. Only 1.21% of the data in the dataset (almost RS. 6 Lakhs) is fraudulent. The response variable, or dependent variable, is 'fraud,' which has a value of 1 if a transaction is fraudulent and 0 otherwise. To adjust the imbalanced dataset, SMOTE [42], [43], [44], [45] is utilized to up-specimen the minority information that is a scam training dataset [46].

5.2 Exploratory Data Analysis

The dataset is highly imbalanced, as shown in Figure 8. Just 7200 (1.21%) is the number of fraudulent transactions, and on the other hand, the amount of authentic transactions is 587443.

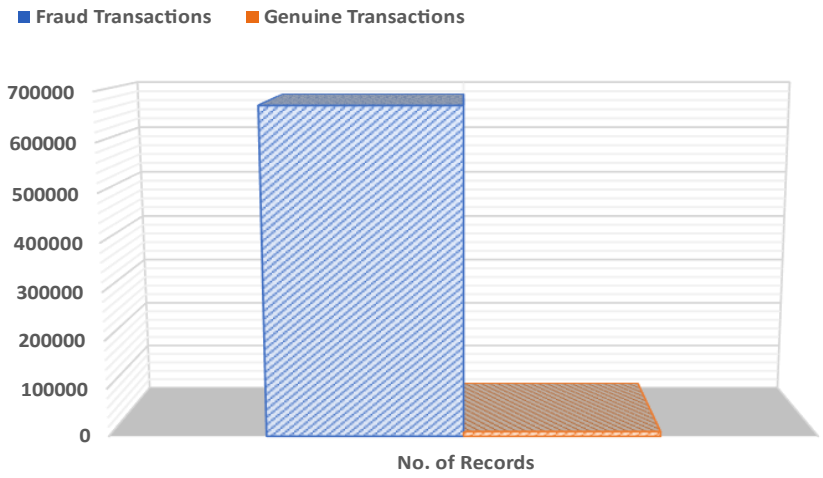

Fig.8: Count of fraud and non-fraud payment

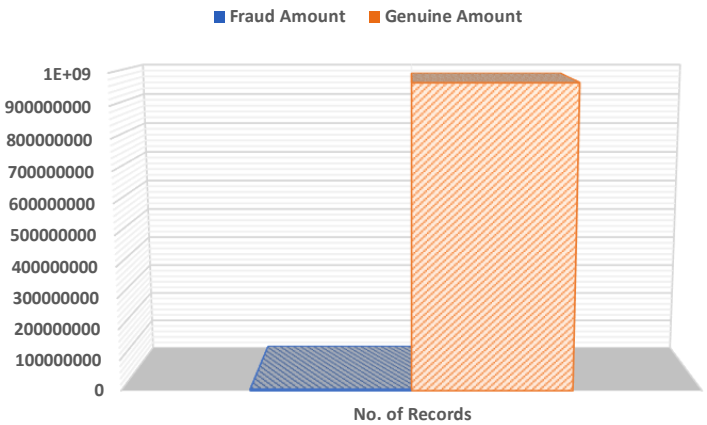

Fig. 9: Fraud and non-fraud transaction payment

The percentage of fraudulent transactions is 16.96. Figure 9 is evident that even low fraud records hold bulk transactions though it is fiddled.

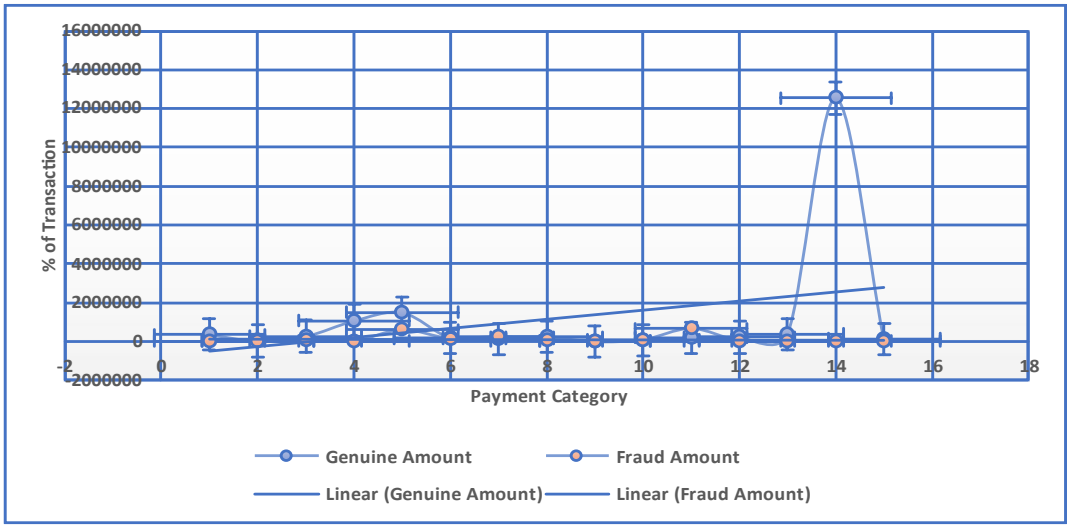

Fig. 10: Category wise fraud and non-fraud transaction payment

We can examine that some classification exchanges are exalted, yet the scam is less and the other way around (Figure 10). Therefore, we may deduce which group is offering important perception outcomes. To evaluate the execution of the model, the composite techniques are correlated. To estimate the performance, the best option is the accuracy and the recollection of false samples especially, in the analysis of slant information.

Figure 11 (a) exhibits the accuracy Recall bend of trial information in the subsequent month 2016.09 for every technique with the irregularity proportion among standard and false examples at 10000:1. The practical scenario shows that NB and IB models are preferable to usual separation models like SVM or LR. Low promotion is given to LB combined NB optimization, and a single LSTM chronological model overcomes high band models. By positioning the ensemble models before or after the LSTM process, a few enhancements can be acquired. Contrastingly, when NB, IB, and LS=TM models are piled in order, unique promotion emerges. In another way, compared to other models, IB→LSTM→NB (C-I-B) model performs better.

Furthermore, as shown in Fig. 11 (b), the NB model's prediction capacity diminishes over time, but the influence of the LSTM model diminishes with some irregular beatings. This indicates that current fraud detection patterns within a single transaction are constantly changing, although sequential patterns may be effective on a regular basis. Nonetheless, it is recommended that all models be trained on a regular basis to ensure that they do not lose their effectiveness.

Indeed, when there is a balance in data, NB model is improved than the single LSTM model. At first, the value of the best F1-score is more than the LSTM model, as shown in Figure 11 (c) by dropping very sharply with a growing imbalance ratio. This indicates that to certain point, the LSTM model can lessen the imbalance. This fundamental merit of LSTM is inherited by the WBW model and can provide comparatively outstanding performance in highly imbalanced scenarios. To summarize, there is a relatively better performance of “WBW” sequence learning architecture for our business world that allows only other more accessible structures such as “BW” or “WB”.

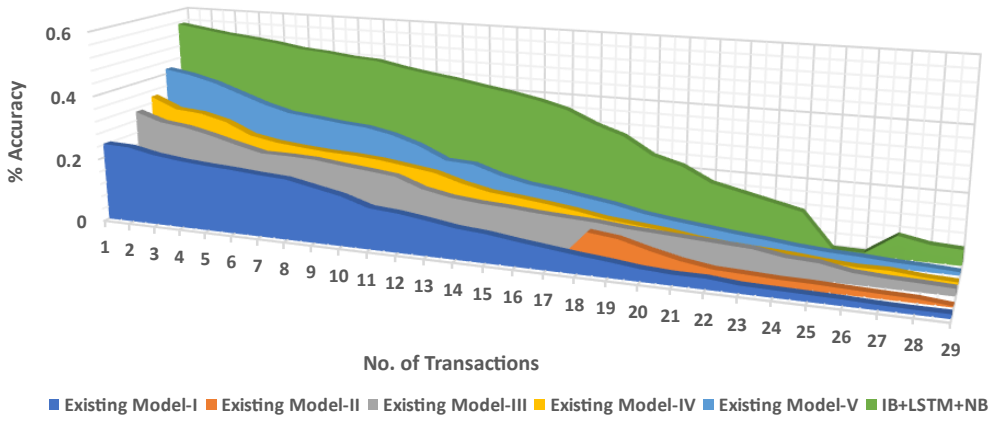

Fig. 11 (a): Comparison of decreasing F1-score trends with increasing imbalance ratio

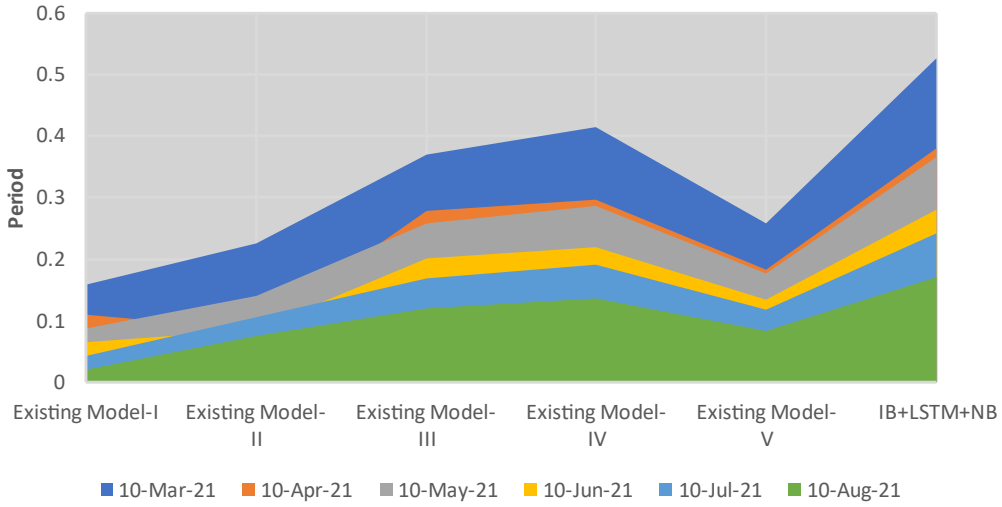

Fig. 11(b): F1-score for WBW attributed to different GRU models

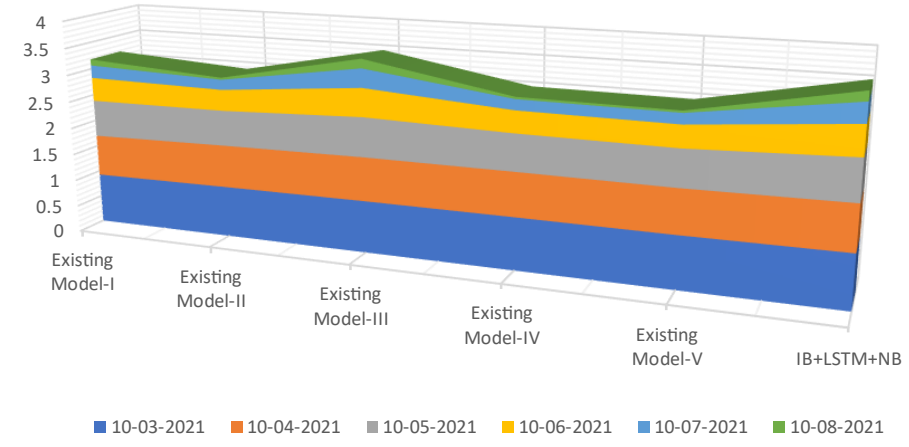

Fig. 11 (c). Results of F1-score with declines and rising imbalance ratios are compared

6.0 CONCLUSION

This article proposes an inclusive assembly method for identifying fraudulent transactions using the Fraud Detection System in collaboration with multiple platforms and algorithms. Combining EM and DL approaches has resulted in a novel composite-structure based DSL architecture known as C-I-B classifier. First, work on AFE is done on Spark. Then, to optimize the characteristics of a single transaction, the I-Bagging model is used. Second, using modified sequential data, the LSTM model is used to develop improved connections between transactions. Finally, by utilizing optimized transaction eigenvectors, a NB classifier is trained. This strategy is more effective than most standard approaches in identifying transaction fraud.

Additionally, the attention process has been used to improve model performance. The whole collaboration model may be combined as a C-I-B sandwich-structured DSL architecture by stacking an EM, an RNN-DL model, and another EM in a particular order. Many additional cases where the information sequence comprises vectors with complicated linked characteristics might benefit from models with comparable architectures.

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] C. Abbye, "*Building a Digital Repository Program with Limited Resources*". Chandos Publishing, ISBN: 978-1-84334-596-1, 2010.

[2] Z. Carson, "*Ten Strategies of a World-Class Cybersecurity Operations Center*". The MITRE Corporation, 2014.

[3] ACFE Report, https://www.acfe.com/report-to-the-nations/2020/docs/infographic-pdfs/Key%20Findings%20from%20the%20Report.pdf, 2020.

[4] L. Joseph et al., "*How to Implement Security Controls for an Information Security Program at CBRN Facilities*", https://www.pnnl.gov/main/publications/external/technical_reports/PNNL-25112.pdf, 2015.

[5] S. Somanchi and R. Telang, Security, "*Fraudulent transactions and Customer Loyalty: A Field Study*". ICIS, 2016.

[6] Deloitte, "*Business Impacts of Machine Learning*". TG_Google Machine Learning Report_Digital Final", 2017.

[7] E. Johannes et al., "*Policy Responses to Fintech: A Cross-Country Overview*" BIS Media and Public Relations, ISSN 2522-2481, 2020.

[8] Ghosh and Reilly, "*Credit Card Fraud Detection with A Neural Network*", Proceedings of the 27th Hawaii International Conference on System Sciences HICSS-94, 1994, Doi:10.1109/HICSS.1994.323314.

[9] R. Brause et al., "*Neural Data Mining for Credit Card Fraud Detection*". Proceedings 11th International Conference on Tools with Artificial Intelligence, doi:10.1109/TAI.1999.809773, 1999.

[10] E. Aleskerov et al., "*CARDWATCH: A Neural Network-based Database Mining System for Credit Card Fraud Detection*", Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering (CIFEr), 1997, doi. 10.1109/CIFER.1997.618940.

[11] G. Tao et al., "*Neural Data Mining for Credit Card Fraud Detection*", International Conference on Machine Learning and Cybernetics, 2008, doi:10.1109/ICMLC.2008.4621035.

[12] M. Syeda et al., "*Parallel Granular Neural Networks for Fast Credit Card Fraud Detection*". IEEE World Congress on Computational Intelligence. IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291). 2002, doi:10.1109/Fuzz.2002.1005055.

[13] A. Dal Pozzolo et al., "*Credit Card Fraud Detection and Concept-Drift Adaptation with Delayed Supervised Information*", Proceedings of International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1–8.

[14] T. Ma et al., "*An Unsupervised Incremental Virtual Learning Method for Financial Fraud Detection*", Proceedings of IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 2019, pp. 1–6.

[15] A. Somasundaram et al., "*Parallel and Incremental Credit Card Fraud Detection Model to Handle Concept Drift and Data Imbalance*", Neural Computing & Applications, Vol. 31, 2019, pp. 3–14.

[16] E. W. T. Ngai, et al., "*The Application of Data Mining Techniques in Financial Fraud Detection: A Classification Framework and An Academic Review of Literature*," Decision Support System, Vol. 50, 2011, pp. 559–569.

[17] M. Ahmed et al., "*A Survey of Anomaly Detection Techniques in Financial Domain*," Future Generation Computer System, Vol. 55, 2016, pp. 278–288.

[18] M. Ahmed et al., "*Anomaly Detection on Big Data in Financial Markets*". In Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Sydney, Australia, 2017, pp. 998–1001.

[19] A. Abdallah et al., "*Fraud Detection System: A Survey*", Journal of Network and Computer Applications, Vol. 68, 2016, pp. 90–113.

[20] K. Gai et al., "*A Survey On FinTech*". J. Netw. Comput. Appl., Vol. 103, 2018, pp. 262–273.

[21] N. F. Ryman Tubb et al., "*How Artificial Intelligence and Machine Learning Research Impacts Payment Card Fraud Detection: A Survey and Industry Benchmark*". Engineering Applications of Artificial Intelligence, Vol. 76, 2018, pp. 130–157.

[22] J. West et al., "*Intelligent Financial Fraud Detection: A Comprehensive Review*". Computers and Security, Vol. 57, 2016, pp. 47–66.

[23] T. Pourhabibi et al., "*Fraud Detection: A Systematic Literature Review of Graph-based Anomaly Detection Approaches*", Decision Support Systems, Vol. 133, 2020, pp. 113303.

[24] A. Srivastava et al., "*Credit Card Fraud Detection Using Hidden Markov Model*". IEEE Transactions on Dependable & Secure Computing, Vol. 5, No. 1, 2008, pp. 37-48.

[25] R. J. Bolton et al., "*Unsupervised Profiling Methods for Fraud Detection*", Proceedings of Credit Scoring & Credit Control, Vol. 7, 2001, pp. 5-7.

[26] Z. C. Lipton et al., "*Critical Review of Recurrent Neural Networks for Sequence Learnin*g", Computer Science, 2015.

[27] B . Wiese and C. Omlin, "*Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks*," 2009.

[28] Li, Xurui et al. "*Transaction Fraud Detection Using GRU-Centered Sandwich-Structured Model*". IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2018, pp. 467-472.

[29] T. Olowookere et al., *"A Framework for Detecting Credit Card Fraud with Cost-Sensitive Meta-Learning Ensemble Approach"*, Scientific African, 2020, e00464.10.1016/J.SCIAF.2020.e00464.

[30] P. Raghavan et al., "*Fraud Detection using Machine Learning and Deep Learning*," 2019, pp. 334-339, doi: 10.1109/ICCIKE47802.2019.9004231.

[31] T. Amarasinghe et al., "*Critical Analysis of Machine Learning-Based Approaches for Fraud Detection in Financial Transactions*". ICMLT'18: Proceedings of International Conference on Machine Learning Technologies, 2018, DOI: 10.1145/3231884.3231894.

[32] S. Shirgave et al., "*A Review On Credit Card Fraud Detection Using Machine Learning*". International Journal of Scientific & Technology Research, Vol. 8, No. 10, 2019, pp. 1217-1220.

[33] K. Kaithekuzhical Leena and C. Ajeet, "*Detection and Prediction of Credit Card Fraud Transactions Using Machine Learning*", International Journal of Engineering Sciences & Research Technology, Vol. 8, No. 3, 2019, pp. 199-208.

[34] S. Maniraj et al., "*Credit Card Fraud Detection using Machine Learning and Data Science*". International Journal of Engineering Research, Vol. 8, No. 9, 2019, pp. 110-115, Doi: 10.17577/IJERTV8IS090031.

[35] R. Blagus and L. Lusa, "*SMOTE for High-Dimensional Class-Imbalanced Data*". BMC Bioinformatics. Vol. 14, No. 06, 2013, pp. 1-16, doi:10.1186/1471-2105-14-106.

[36] G. Tüysüzoğlu and D. Birant, "*Enhanced Bagging (eBagging): A Novel Approach for Ensemble Learning*". The International Arab Journal of Information Technology, Vol. 17, 2020, pp. 515-528, doi:10.34028/iajit/17/4/10.

[37] D. E. Rumelhart et al., "*Learning Representations by Back-Propagating Errors*", Nature, Vol. 323, 1986, pp. 533–536.

[38] P. J. Werbos, "*Generalization of backpropagation with Application to A Recurrent Gas Market Model*", Neural Networks, Vol. 1, 1988, pp. 339–356.

[39] J. L. Elman, "*Finding Structure in Time*", Cogn. Sci., 14, 1990, pp. 179–211.

[40] S. Hochreiter et al., "*Long Short-Term Memory*". Neural Computation, Vol. 9, 1997, pp. 1735–1780, doi:10.1162/neco.1997.9.8.1735.

[41] C. Olah, "*Understanding LSTM Networks*", http://colah.github.io/posts/2015-08- Understanding-LSTMs, 1999.

[42] R. Blagus et al., "*SMOTE for High-Dimensional Class-Imbalanced Data*", BMC Bioinformatics, Vol. 14, No. 106, 2013, pp. 1-16, doi: 10.1186/1471-2105-14-106.

[43] 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, doi: 10.22452/mjcs.vol34no3.1.

[44] 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, doi:0.22452/mjcs.sp2020no1.5

[45] 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, doi: 10.22452/mjcs.vol34no3.5.

[46] 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.