I.J. Information Engineering and Electronic Business, **2024, 6, 1-17**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2024.06.01

Suspicious Activity Detection Model in Bank Transactions Using Deep Learning with Fog Computing Infrastructure

Girish Wali*
SVP - Business Architecture, IT Business Lead Analyst, Business Analysis & Product Solutions at CITI Bank, India
Email: waligirish@gmail.com
ORCID ID: https://orcid.org/0009-0006-3213-3168
*Corresponding Author

Chetan Bulla
Professor of Computer Science and Engineering, India
Email: chetan.bulla@syenchron.com
ORCID ID: https://orcid.org/0000-0002-6463-0509

Received: 07 February, 2024; Revised: 22 May, 2024; Accepted: 24 June, 2024; Published: 08 December, 2024

Abstract: The financial sector is grappling with significant challenges in detecting cyber attacks, leading to potential short- and long-term financial losses for banks and other institutions. The statistical and machine learning methods have been effective in identifying suspicious activities, they have struggled to achieve a balance between recall and precision. To improve accuracy, this paper introduces a novel approach that employs deep learning and bio-inspired algorithms to detect suspicious activities. The proposed model analyzes transactional patterns, quantities, and temporal aspects using a carefully curated dataset of labeled transactions. The model shows promising results in distinguishing between legitimate and fraudulent operations, achieving a balance between recall and precision. Further, many in the industry are transitioning to cloud computing infrastructures to enhance application performance. However, these infrastructures are not ideal for delay-sensitive applications, such as those in the medical and finance sectors. To address communication delays, fog computing has emerged as a new paradigm. The proposed model was simulated using Python and the Google Colab framework, and experimental results shows that improved accuracy and a balanced recall and precision.

Index Terms: Suspicious activity detection, Deep learning model, LSTM, Bio-inspired algorithms, Fog Computing, Google Colab

1. Introduction

The rapid growth of transactions in todays digital financial world has sparked the emergence of more sophisticated tactics for committing banking frauds [1]. It is essential to improve the security and authenticity of bank transactions by adopting advanced detection models to identify the unauthorized and suspicious activities. The existing statistical mechanisms static in nature and do not support dynamically changing domains. Further, a new era of machine learning and deep learning model captures the dynamism but these models needs appropriate optimization technical that support dynamism and increase the performance. So it is essential to use appropriate optimization methods to improve the accuracy. This paper proposed a novel solution where it combines deep learning model and bio-inspired algorithm to improve the accuracy of suspicious activity detection model.

When it comes to digital transactions frequent use poses a challenge because of the nuanced differences in fraudulent behavior necessitating a more sophisticated and flexible strategy Deep learning methods provide a hopeful resolution for swiftly and precisely pinpoint suspect activities in banking transactions by leveraging their capacity to identify intricate patterns and connections, within extensive datasets. Numerous financial institutions are switching from in house computing setups to cloud computing because of the many advantages it offers. Like scalability and reliability as well as a pay as you go approach with robust support [2]. The cloud computing may not be the fit for time sensitive applications such as medical or financial systems due, to the communication delays when the cloud data centers are far away from stake holders. To mitigate this communication delays and enhance performance cloud service, a new data-

centers need to install near to stake-holder locations and it is practically not possible due to hardware / firmware and maintenance cost. A new computing paradigm called Fog Computing, was introduced to avoid the communication delay where computational devices are installed closer to data sources [3], thereby reducing communication delays. These computational devices are lightweight and it need less investments. Figure 1 Shows the fog computing infrastructure for financial and banking applications.

The increasing frequency of cyberattacks and the cleverness in fraudulent activities impose a shift from traditional Models [5]. The deep learning is aa subset of machine learning [4], that mimics the human brain (artificial neural network) has proven records to achieve good accuracy and performance for different types of applications. These models provide a practical and intelligent protection against malicious activities by utilizing the Long Short-Term Memory (LSTM) neural network model [6], which enables autonomous learning and adaptation to embryonic patterns of suspicious activity behavior. This research work combines the new intelligent algorithm to improve the accuracy in detecting suspicious bank transactions.

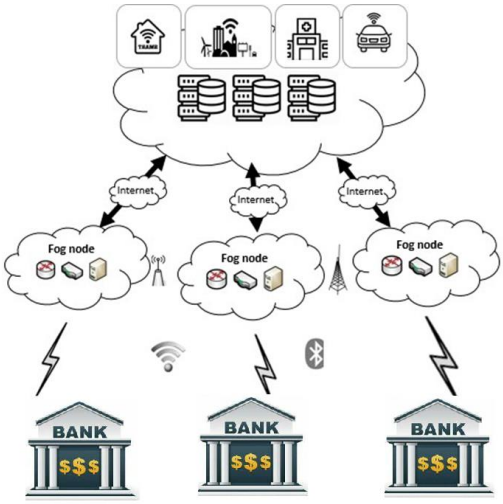

Fig. 1. Adoption of fog computing in the financial sector

The proposed model uses the LSTM deep learning model [6] to detect suspicious activities in bank transactions, while also combining a bio-inspired algorithm called the Artificial Bee Colony to optimize the LSTM model's performance. The data preprocessing is necessary for the improving performance of machine learning and deep learning models. Converting raw data into a curated form is a challenging task as it contains duplicate, null, invalid, inconsistent data and outlier data. The proposed model uses various data preprocessing methods to standardize the data by converting raw data to curated data. LSTM, a deep learning model based on the Recurrent Neural Network (RNN), is particularly suited for analyzing temporal data, such as time-series data, to detect suspicious activities and identify patterns. One of the main advantages of LSTM is to overcome the vanishing gradient problem, allowing it to learn from inputs that are independent of each other and retain longer memories. This gives more accurate and effective solutions in dynamically and complex patterns of data. To further optimize the LSTM model's performance, a bio-inspired algorithm [7,12] is used to select the optimize the values of hyperparameters, achieving a good balance between recall and precision.

The scope of this research article encompasses the development and evaluation of a Suspicious Activity Detection Model for bank transactions, using deep learning techniques within a fog computing infrastructure. The study focuses on designing a robust model capable of identifying suspicious transaction patterns to avoid fraudulent or suspicious activity in banking sectors. Key components of the research include the integration of deep learning algorithms to enhance the accuracy of detection, and the utilization of fog computing to efficiently process and analyze transaction data at the edge of the network. The research aims to address challenges associated with real-time detection, detection accuracy and latency, providing insights into how fog computing can optimize the performance of deep learning models in financial domains.

The combination of a Long Short-Term Memory (LSTM) model with an Artificial Bee Colony (ABC) bio-inspired algorithm is highly effective for detecting suspicious activities in banking systems due to the distinct but complementary strengths of each method in managing complex data patterns and optimization tasks. Here's why this combination is particularly advantageous:

1. The LSTM model for looking at sequential data:The LSTM model is a type of recurrent neural network (RNN) that is built to handle sequential data, which is popular in banking. The time and order of transactions can often show important trends that can point to fraud or other bad behavior. One of the best things about LSTM networks is that they can find long-term connections and ties in time series data. This is very important for finding strange things that might not be clear from looking at short-term data but become clear when you look at long-term chains of transactions. Ability

to Respond to Changes in Behavior: LSTM models can adapt quickly to changes in how users act, which makes them very good at finding new fraud schemes that static models might miss.

2. The ABC algorithm for improving model performance: The ABC algorithm uses the way honey bees find food, which is known for being a fast and effective way to find the best answer in large search areas. In the context of finding fraud, this method can be used to improve the LSTM model's settings, making it better at finding oddities. The ABC algorithm is very good at exploring and taking advantage of search spaces. This lets it fine-tune the LSTM model's hyperparameters, like the learning rate, the number of layers, and the size of each layer. This makes the model more correct and trustworthy. The ABC method is flexible and can be used to solve many different kinds of optimization problems, even ones that aren't linear or have more than one goal. This flexibility is very important in banking systems where what is considered suspicious behavior can change a lot and may need various optimization techniques.

3. Synergistic Integration: The LSTM model's ability to understand and predict sequential data patterns and the ABC algorithm's ability to find the best model parameters work together to make a powerful tool for finding fishy activities in bank transactions.Also, the ABC algorithm's improvement of the LSTM model makes it easier to tell the difference between normal and suspicious transactions. This lowers the chance of false positives and improves the total accuracy of identification. When the ABC algorithm is used to fine-tune an improved LSTM model, it can work in real time. This makes it perfect for constantly watching banking systems where finding scam quickly is important..

The main objective of proposed model are as follows:

1. Integrating deep learning models with fog computing infrastructure greatly improves the precision of identifying suspicious activity in bank transactions when compared to conventional centralized solutions. It is possible to achieve enhanced real-time analysis and detection of fraudulent actions by integrating deep learning algorithms with the distributed architecture of fog computing, resulting in improved outcome accuracy.
2. The implementation of fog computing infrastructure decreases the delay in identifying questionable transactions in banking systems, allowing for quicker response times in comparison to alternatives that only rely on cloud technology. Fog computing aims to decentralize processing power by placing it in close proximity to the data source, such as ATMs and bank offices. Processing data locally in the fog will decrease the distance it needs to travel, leading to faster identification and mitigation of suspicious behaviors.
3. The deep learning model is capable of accurately detecting growing patterns of suspicious activity that go unnoticed by rule-based or standard machine learning methods. The Long Short Term Memory (LSTM) model is proficient at acquiring intricate patterns from extensive datasets. Therefore, the model has the ability to detect novel forms of fraudulent activity that traditional approaches may overlook.

1.1 Scope of research work

This research study focuses on the creation and assessment of a Suspicious Activity Detection Model for bank transactions. The model utilizes advanced deep learning techniques and is implemented within a fog computing infrastructure. The project aims to create a resilient model that can accurately detect and highlight abnormal transaction patterns that suggest fraudulent or suspicious behavior. The research incorporates deep learning methods to improve the accuracy of detection and utilizes fog computing to quickly process and analyze transaction data at the network's edge. The research seeks to tackle the difficulties related to detecting and scaling in real-time, offering valuable information on how fog computing might enhance the efficiency of deep learning models in financial settings. The scope of the project includes assessing the performance of the model using established measures such as accuracy, recall, and F1-score. Additionally, an examination of the model's sensitivity and specificity will be conducted.

1.2 Outline

The significance of the proposed model lies in its potential to transform the financial security landscape by delving into the intricacies of deep learning-based detection of suspicious activities. Financial institutions can leverage neural network-based models not only to detect and respond swiftly to known fraudulent behaviors but also to identify and address emerging risks. The remainder of the paper is organized as follows: Section 2 reviews the current state of research related to suspicious activity detection. Section 3 details the proposed model and its underlying principles. The experiment and its evaluation are discussed in Section 4. Finally, Section 6 presents the conclusion and offers recommendations based on the research findings.

2. Literature Review

Logistic regression is a type of machine learning that is used to guess the values from a binary classification. The explanatory factors don't have to be related or have a normal distribution for this method to work [16]. The results of Logistic Regression models are qualitative, and the factors used to explain them can be numbers or groups. A lot of experts have used logistic regression to figure out which banks are going bankrupt. In banking, decision trees can be used for many things, like finding different types of risk based on patterns, finding transactions that are likely to go wrong, and sorting transactions into those that are legal and those that are not. It is suggested that a data driven

multiagent-based anomaly detection model be used to find changes in the sensors that are set up in a fog-based cloud computing system [14]. In the process of anomaly identification, different bots are placed in different places and used to carry out different tasks. The LSTM deep learning method is used to find sensor readings that don't seem to match up. The results of the experiment show that the proposed model is more accurate than the most recent cutting-edge study on anomaly detection.

Decision trees are a type of non-linear classification that uses a set of factors to split a sample into smaller and smaller subgroups. According to a set of rules, the process picks the explanatory variable that is most strongly linked to the outcome variable at each branch of the tree [17]. It's easy to add a variety of quantitative or qualitative data structures because it's not parametric, and you don't even have to choose training data. But when decision trees are used on the whole set of data, they tend to overfit the training data, which could lead to bad results. The R random forest package [19] was used to make models for bagging and random forests. It is possible to rate how important each trait is compared to the training set. Random forests, on the other hand, prefer traits that have a lot of levels for data, like qualitative variables that have a lot of levels. Random forest can be used to separate parts of movies, sort pictures for pixel analysis, and look at big biological data in biology. There needs to be a safe and effective way to keep track of the details given by the agent. The proposed model [15] is the fastest, safest, and most useful way to find fraud in this study. In the BCPS for the post-quantum era, cognitive computing and quantum computing were used to find strange entities. The suggested way correctly labels transactions as either real or fake 97.04% of the time and makes only 0.03% of mistakes.

A novel technique for detecting fraudulent financial transfers by utilizing actual bank data and assessing client information, transactional actions, and customer connections is introduced in [18]. In order to tackle the problem of limited fraudulent data samples and minimize training challenges arising from imbalanced data, different advanced generative models such as the conditional variational autoencoder and Wasserstein generative adversarial network are utilized to produce extra fraudulent raw data and enhance the number of fraud samples in the latent space. The reparameterization approach is used into the encoder-decoder architecture to enhance the model's generative capacities. Moreover, a Graph Neural Network (GNN) is employed to represent customer connections. The suggested method employs end-to-end learning by combining the reconstruction loss of the autoencoder, the KL divergence loss (when the reparameterization strategy is used), and the classification loss for fraud detection. In order to enhance the use of computing resources, the technique of neighborhood sampling for Graph Neural Networks (GNN) is integrated with mini-batch training for the autoencoder. This integration leads to improvements in both the efficiency of the training process and the dependability of the resulting model. The efficiency of the proposed fusion network is demonstrated through comprehensive experiments, which emphasize the significance of each component and preprocessing phase. The model demonstrates significant enhancements in fraud detection, as seen by the increased areas under the precision-recall curves.This approach shows promise for broader applications in anomaly identification, especially in datasets with irregular time series and intricate customer interactions. The model improves 2% accuracy as compared to existing models.

[22] made a machine learning-based way to find credit card fraud by using hybrid models with Ada Boost and majority vote processes. For ease of use, they added noise to their blend models at levels between 10% and 30%. Multiple vote procedures got a good score of 0.942 based on data from the sample with 30% more noise. In the end, they decided that the vote system was the best way to deal with the noise. [23] offered two different random forest models that were used to teach the behaviors of both normal and abnormal trades. of these two different kinds of random woods.

The asynchronous advantage actor-critic (A3C) method [19] is the most advanced and dynamic reinforcement learning system used to solve the problem of finding suspicious behavior. The experiment shows that the results are better at recalling and identifying strange behavior compared to other machine learning-based models. A vector convolution deep learning (VDCL) model is used to find big differences in the Internet of Things. The main goal is to make it easier to identify suspicious behavior on a larger scale, since network traffic records in the Internet of Things can't naturally be scaled up. The VCDL model helps the master fog node sort known and unknown problems into groups. The experiment results showed that the suggested framework was more accurate, precise, and able to remember things compared to the current method for finding suspicious behavior.

The research emphasizes the integration of computer vision and artificial intelligence in automated video detection systems. This combination is essential for enhancing the effectiveness of surveillance in public spaces, addressing the limitations of traditional method [24]. One of the primary functionalities of the proposed system is its ability to detect anomalies in surveillance footage using CNN model. By differentiating between normal and abnormal behaviors, the system can effectively flag potential security threats, thereby enhancing public safety. The accuracy of model is not encouraging and but optimization in multiple stages.

The paper introduces a novel model that integrates Long-term Recurrent Convolutional Networks (LRCN), combining the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks [30]. This integration allows for effective learning of visual features from video frames, enhancing the model's ability to recognize and classify behaviors accurately. The model is implemented using widely-used frameworks such as Python, Keras, and TensorFlow and recorded good accuracy of approximately 91.55%. Although the model can classify 13 different categories of suspicious activities, this may not encompass all possible behaviors that could be deemed suspicious. While the model is designed to function with limited resources, the initial training phase may still require

significant computational power and time, especially when dealing with large datasets. The authors introduce a novel approach for detecting fraudulent financial statements (FFS) by utilizing an ensemble model that combines Feedforward Neural Networks (FNNs) and Long Short-Term Memory (LSTM) networks [31]. This method enhances the detection of complex fraud patterns that traditional techniques may overlook. The Spotted Hyena Optimizer, a metaheuristic optimization method inspired by spotted hyena social behavior, is a major contribution. Designed to optimize LSTM model weights and biases, this optimizer improves fraud detection. The Spotted Hyena Optimizer, in conjunction with the ensemble model that combines FNNs and LSTMs, may introduce complexity that could impede its practical implementation in real-world scenarios.

Table 1. Related works on suspicious activities in bank / finance sectors

Ref	Methodology	Advantages	Limitations
[14]	LSTM Deep learning Model	Good Accuracy	Limited Dataset, more complexity
[15]	Quantum Computing	Good Accuracy, Time Effective	More Complexity
[16]	Logistic Regression	Moderate Accuracy	Limited Dataset features, moderate complexity
[17]	Decision Tree	Moderate Accuract	High Complexity, More training time
[18]	Graph Neural Network	High Accuracy	High Computation resources consumption
[19]	Random Forest	High Accuracy	High Computation resources consumption
[21]	Hidden Markov Model	Moderate Accuracy	Static in Nature, traditional Method
[22]	hybrid models with Ada Boost and majority vote processes	High Accuracy	High Complexity, High Computation resources consumption
[23]	Random Forest	High Accuracy	More complexity as it uses two RF model to train normal and suspicious activities.
[24]	CNN	Moderate Accuracy	Video analysis
[25]	Support Vector Machine model	Moderate Accuracy	High Complexity
[26]	Auto-encoder	Moderate Accuracy	Takes more training time when dataset features are changed
[30]	LSTM	Good Accuracy	Performance can be improved if optimizations were used
[31]	FF LSTM	Good accuracy	Performance can be improved if optimizations were used
[34]	LSTM	Goof Accuracy	High Complexity

The existing model have multiple limitations, first, complexity increases as accuracy increases, it is essential to keep tradeoff between accuracy and complexity. Second, latency is compromised when cloud based services are used. Third, the features in the dataset are limited, due to this accuracy will increase, but in real time, more number of features are used in real world applications. Table 2 shows related work with its limitations. In this paper, the LSTM model is used to detect suspicious activity in the present transaction based historical data. The LSTM extract the features from historical data and categorized binary classification. The present transaction data matches with extracts features then that transaction is considered as suspicious the paper contribution to knowledge includes:

- Design and Develop a deep learning model called LSTM to detect suspicious activity in the transactions.
- Preprocess the dataset using standard techniques normalization, and filling missing and wrong values
- Tune the hyperparameters of LSTM model to improve the accuracy of suspicious transactions using bio-inspired algorithm.
- Integrate the deep learning model in fog computing infrastructure to reduce the communication delay and get the model result quickly.

3. Methodology

The proposed model for detecting suspicious activity is structured into three key processes: (i) Data ingestion, (ii) Data preprocessing, and (iii) Suspicious activity detection. The input dataset comprises historical data, which includes both suspicious and non-suspicious transactions. This dataset contains detailed information about past transactions as well as current ones. Figure 2 illustrates the different stages of the proposed model aimed at detecting suspicious activity. Data is ingested into the model through two primary methods: historical load and incremental load. These transactional data sets are then fed into a deep learning model, which is trained and tested to identify suspicious activities by analyzing patterns in the historical data. Accurate identification of suspicious transactions heavily depends on the quality of the data. Therefore, a data preprocessing mechanism is employed to refine the dataset by addressing issues such as missing values, outliers, and null entries. To enhance the accuracy of the Long Short-Term Memory (LSTM) learning model, a bio-inspired algorithm is utilized to optimize performance by selecting suitable hyperparameters. The process of suspicious activity detection and performance tuning is iterative, continuing until the model achieves the highest possible accuracy.

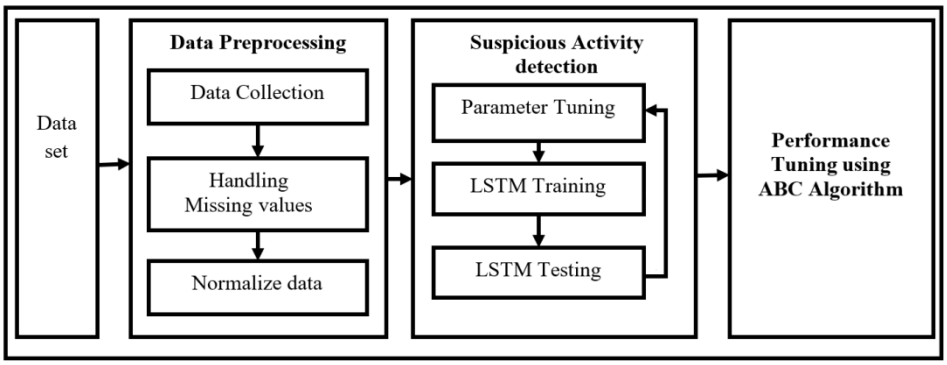

Fig. 2. Working principle of suspicious activity detection

3.1 Data ingestion

Data is collected from several streaming servers and stored in a designated landing zone. There are two primary types of data loading processes involved: the initial load and the incremental load [9]. The initial load refers to the process of transferring all historical data into the system for the first time. In contrast, the incremental load involves regularly updating the system with new data at predefined intervals, which could be hourly, daily, weekly, or monthly. Since the model is designed for bank transactions, we have configured the data loading process to occur daily. This ensures that all transactions are processed at the end of each day, keeping the system up to date with the latest information. Figure 3 shows the process data ingestion

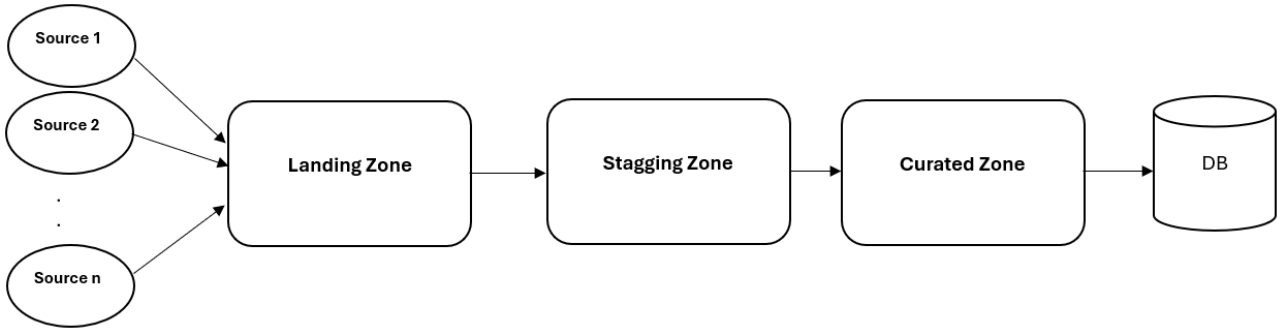

Fig. 3. Data ingestion Process

3.2 Data Preprocessing

When training deep learning models, ensuring high-quality data is crucial for achieving accurate and reliable results. The process of refining and preparing data, often referred to as data preparation, involves eliminating irrelevant or redundant information and transforming the data into a suitable format for deep learning algorithms. In the proposed system, data is collected, filtered, and normalized from a transaction dataset using a streaming server. This data preparation involves several steps, including data collection, aggregation, filtering, handling missing values, and normalization, all of which are facilitated by a fog monitoring system.

The quality of data is one of the key factors that determine the performance of deep learning models. To enhance this quality, the proposed model employs three main data processing strategies: addressing missing data and normalizing the data. Missing data is handled by calculating the mean, median, or mode of a specific feature and using these values to fill in the gaps. Additionally, to avoid the redundancy of collecting similar data multiple times, the model is designed to gather temporal data only when there is a significant difference between past and current data. This ensures that the data used for training the model is both relevant and diverse, which contributes to the model's overall effectiveness.

$$CD_t = \frac{D(t)-D(t-1)}{MAX} X\ 100 \tag{1}$$

The training data's missing values are filled in by using the feature's mean and median values

$$\underline{x} = \frac{\sum_{i=1}^{N} x}{N} \tag{2}$$

$$Median = \left(\frac{n+1}{2}\right)^{th} term \quad \text{(if n is even)} \tag{3}$$

$$Median = \frac{\left(\frac{n}{2}\right)^{th} term + \left(\frac{n}{2}+1\right)^{th} term}{2} \quad \text{(If n is odd)} \tag{4}$$

Addressing outliers in data is essential for improving the performance of deep learning models. Outliers are data points that significantly differ from the majority of the dataset, and they can skew the results and hinder the model's ability to learn effectively. To mitigate this impact, it's important to normalize the data by scaling it to a specific range, such as between a minimum and maximum value. This process helps ensure that all data points are treated consistently, thereby enhancing the accuracy and reliability of the model's predictions.

$$X' = \frac{x - min(x)}{max(x) - min(x)} \tag{5}$$

To detect suspicious activity, a Long Short-Term Memory (LSTM) network model is trained using processed data to enhance accuracy. The proposed approach leverages historical data trends to identify potentially suspicious behaviors. Various features are extracted from this historical data, enabling the model to learn patterns associated with suspicious activities. These learned patterns are then compared against current transaction patterns to detect anomalies. In time-series data, certain patterns may emerge, and when temporal data interference is present, such patterns can indicate suspicious activities.

To effectively model time-series data, an LSTM, a type of Recurrent Neural Network (RNN), is particularly well-suited. LSTM networks are advantageous over traditional RNNs because they can retain information over long sequences, learn from inputs that may not be directly related, and address the vanishing gradient problem commonly encountered in RNNs. Time-series data often include unexpected gaps between critical events, making LSTM networks ideal for analyzing, classifying, and predicting based on such data.For the task of detecting suspicious activity, the LSTM network model is utilized. The architecture of the LSTM unit, which includes components specifically designed for managing long-term dependencies and sequence prediction, is illustrated in Figure 4.

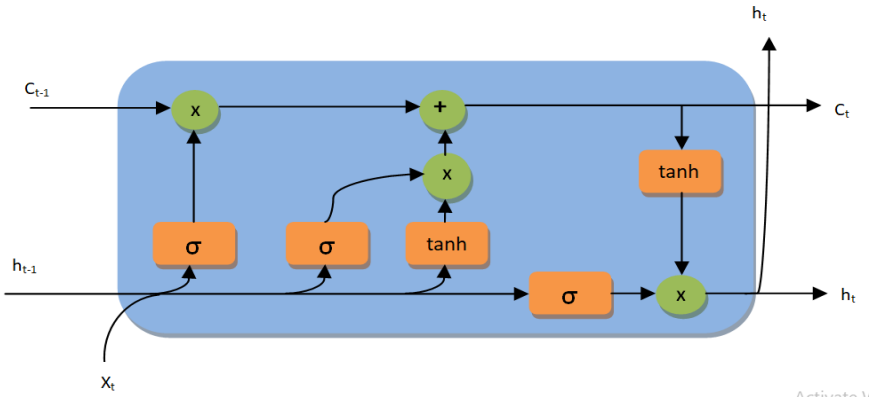

Fig. 4. LSTM Model

The Long Short-Term Memory (LSTM) network operates using three key gates: the input gate, the forget gate, and the output gate. These gates play a crucial role in regulating the flow of information within the LSTM cell. The input gate selectively permits only relevant information to enter the LSTM cell, effectively filtering out unnecessary input. Meanwhile, the forget gate discards any data deemed irrelevant as it enters the cell, ensuring that only valuable information is retained. Finally, the output gate controls which values are transmitted as the output of the LSTM cell.

At a specific time step t, the input X_t is fed into the network. The forget gate assesses the significance of the output from the previous time step $h_{t-1,}$ in the context of the current input X_t Based on this assessment, the forget gate determines which portions of the information should be preserved in the cell state. To make this decision, the activation function is applied to the sum of $h_{t-1,}$ and X_t , which helps to evaluate the importance of the preceding value. If the result of the activation function is close to zero, the information is considered unimportant and is discarded; otherwise, it is retained. The operation of the forget gate is governed by the following process:

$$f_t = \sigma(Wt_f . [h_{t-1,} X_t] + bs_f \tag{6}$$

In this context, σ represents the sigmoid activation function, "." signifies matrix multiplication, and W_{tf} are the weights associated with the forget gate, while b_{sf} denotes the corresponding bias. The next phase includes two key

processes that determine what new information should be added to the cell state. Initially, the input gate evaluates which portions of the cell state need to be updated. Following this, the *tanh* activation function is applied to produce a vector containing the potential new values that could be incorporated into the cell state.

$$i_t = \sigma(Wt_i.[h_{t-1,}X_t] + bs_i \quad (7)$$

$$\sim C_t = tanh(Wt_c.[h_{t-1,}X_t] + bs_c \quad (8)$$

The current cell state C_t is then modified using the outputs from the input gate, forget gate, and *tanh* layer:

$$C_t = f_t * C_{t-1} + i_t * \sim C_t \quad (9)$$

Finally, the output gate and a *tanh* function compute the network's output, as

$$O_t = \sigma(Wt_o.[h_{t-1}, x_t] + bs_o) \quad (10)$$

$$h_t = O_t * tanh(C_t) \quad (11)$$

The terms (Wt_o, bs_o) represent the bias and input weight for the output gate, respectively. To mitigate the risk of overfitting, dropout regularization is employed to randomly deactivate certain neurons during training. This technique involves training each hidden unit with a randomly selected subset of other units, which helps prevent overfitting by ensuring that no single neuron becomes overly reliant on its neighbors. By dropping out a fraction of hidden units and their connections during each training iteration, dropout encourages each unit to develop more robust and independent features. This process reduces the likelihood of overfitting, as the model becomes less dependent on specific neurons and learns to generalize better. Dropout is applied exclusively to the hidden layers in the proposed model.

The underlying idea is to train an LSTM-based neural network to predict the next "n" time steps using regular data inputs. If the difference between the predicted and actual values exceeds a predetermined threshold, the data is classified as anomalous. Anomalies are then categorized into three classes: warning, error, and issue, based on the density of these anomalous points.

The LSTM (Long Short-Term Memory) model plays a crucial role in developing a system designed to detect suspicious activities. In this context, consider $X = \{x1, x2... xt-1\}$, as a sequence of time series data, where each xix_ixi denotes the value at time-stamp iii. The objective is to identify any anomalous or suspicious behavior associated with the transaction occurring at x_t.

To achieve accurate multi-step suspicious activity detection, a sliding window technique is employed. This technique involves analyzing a set of historical data points within a defined window size www. Specifically, the sliding window encompasses values from $\{xt-w, xt+1-w...xt-1\}$, allowing the model to utilize recent historical data for detecting anomalies.

The LSTM network is then trained using these sequences in mini-batches over nnn epochs. This iterative training process helps the model learn and adapt to patterns of normal and suspicious behavior, improving its ability to flag potential anomalies effectively.

$$y_t = O_n(y_t, y_{t+1}, y_{t+2} \ldots . y_{t+n-1}) \quad (12)$$

Where "y_t " is the suspicious activity, O_n is any LSTM Learning model outcome.

3.3 Hyperparameter optimization

The Artificial Bee Colony (ABC) algorithm is inspired by the foraging behavior of bees, mimicking their efficient search for food sources. This bio-inspired approach has been effectively applied across various fields to enhance performance. The ABC algorithm classifies bee colonies into three types: employee bees, bystander bees, and scout bees. Employee bees use their memory to locate nearby food sources and share this information with others. Bystander bees, leveraging this shared knowledge, can then identify high-quality nectar sources. Scout bees, after finding new food sources, transition into employee bees. To improve accuracy, the proposed model employs the ABC algorithm to fine-tune its hyperparameters.

In machine learning, hyperparameters control the training process and can be optimized to enhance model performance. The goal of using ABC optimization strategies is to determine the best hyperparameter configuration to improve the detection accuracy of suspicious activities. Automatic hyperparameter optimization, facilitated by the ABC method, reduces manual effort and computational resources compared to traditional manual tuning, which is labor-intensive and resource-heavy. The effectiveness of the hyperparameter settings depends on the nature and complexity of the problem at hand. By identifying the optimal hyperparameter values for a specific application, the machine learning model's performance is significantly improved, leading to more consistent and reliable results.

In the context of suspicious activity detection, the ABC method is used to optimize the Long Short-Term Memory (LSTM) network, aiming to enhance detection accuracy. The relevant hyperparameters for the LSTM model are selected through the ABC algorithm. This algorithm, inspired by the bees' behavior of storing high-quality honey, involves using 'n' bees to evaluate 'n' food sources, with observer bees collecting data on nectar quality from employed bees. The LSTM model benefits from this optimization by refining hyperparameters to maximize accuracy in detecting suspicious activities over multiple steps. To address challenges related to exploration and exploitation in the conventional ABC algorithm, modifications have been made to improve search efficiency and reduce the time required to find optimal solutions.

1. Predefined values, not arbitrary integers, make up the initial hyperparameters. These numbers are derived from professional advice and provide the initial hyperparameter optimum values. As an illustration: Rate of Learning: 0.1
2. To reduce search space and time, a new search method is implemented. We'll search between *pbest* and *gbest* for the solution. Therefore, the following tactic is used to enhance the search method::

$$v_{ij} = x_{ij} + \varphi(x_{ij} - x_{kj}) + \psi_{ij}\left[\left(\frac{p_{Best}+g_{best}}{2} - x_{ij}\right) + \left(\frac{p_{Best}+g_{best}}{2} - x_{ij}\right)\right] \quad (13)$$

Where φ(x_ij-x_kj) is the difference between the current and new locations, ψ_ij is a random value between [0,1], and Xiji is the current position. The third and fourth terms of the aforementioned equation look for the local best and global best answer. Figure 5 shows the algorithm for proposed model.

Algorithm: Suspicous activity detection with hyperparameter optimization using Artificial Bee Colony

1: Initialize the Hyperparameter value (min, max) values, window size, time-steps, Employee Bees (EB), Onlooker Bees(OB), Maximum Cycle Number MCN), Solutions (SN), k=0, and set the threshold value (0.5)

2. Create Initial population HP_x={SLW, N, DR, LR, RR, OP} and assign random values

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min})$$

Where *i* = 1. . . SN, *j* = 1. . . D. SN is the number of solutions, and D is the number of optimized hyperparameters.

3. Read input values to sliding windows
4. Initialize the counter= 0
5. ***do***

Counter++
Assign sliding window values to LSTM network
Train the LSTM network
Evaluate the fitness value of training using

$$fitness_i = \begin{cases} \frac{1}{(1+f_i)} \; if \; f_i \geq 0 \\ 1 + abs(f_i) if \; f_i < 0 \end{cases}$$

$$p_i = \frac{fitness_i}{\sum_i^{SN} fitness_i} \quad \text{for i = 1.. EB}$$

Check fitness values previous solution
if (p_i<threshold) then
break
else
create a new solution by modifying old solution (hyperparameters)

$$v_{ij} = x_{ij} + \emptyset_{ij}\,(x_{ij} - x_{kj})$$

While(counter >=MCN)

6. Memorize the best hyperparameter combination.
7. Get multi-step prediction values from trained LSTM model
8. calculate the error vector and standard deviation
9. do
if SD >= threshold then
k++
while (timesteps)
10. if k>1 then
Print " Suspicious activity"

Fig. 5. Algorithm for suspicious activity detection

4. Results and Discussion

In this section, we will examine the outcomes of implementing our model for detecting suspicious activities in bank transactions. This model employs advanced deep learning techniques in conjunction with fog computing infrastructure to identify potentially fraudulent behavior more effectively. It provides a comprehensive overview of the experimental setup, dataset specifications, and performance evaluation for the proposed model designed to detect

suspicious activities. Additionally, it includes a comparative analysis between the suggested model and existing state-of-the-art techniques for detecting suspicious behavior.

4.1 Simulation Setup

The simulation model was developed using TensorFlow Keras in conjunction with the Python programming language. To conduct the evaluation of this model, we utilized a Google Colab environment equipped with a 2 GB GPU, 16 GB of RAM, and 160 GB of storage. Python served as the foundational language for implementing various components of the model, including data ingestion, preprocessing, and the detection of suspicious activities. This was achieved by leveraging several predefined libraries to streamline the development process.

4.2 Dataset

The dataset used for customer transactions includes a variety of attributes, such as is_Alerted, is_Suspicious, Account_number, Transaction_date, Transaction_location, Transaction_amount, correspondent_bank, party_name, primary_party_account_number, debit_credit, Account_First_name, Account_Last_Name, Account_Open_date, Account_type, Account_Classification, Risk_level, Zipcode, Annual_income, citizenship_country, email, address, branch, Business_Sector, identification_number, Score1 through Score7, Score, is_noncitizen, and so forth. The Score column is computed using a set of predefined rules, including thresholds like a transaction amount greater than $10,000 and other risk-related criteria that are part of the ETL (Extract, Transform, Load) process. The Risk_level also influences the score calculation, alongside various other rule-based scenarios typical of ETL logic. The is_Alerted column indicates transactions flagged by the ETL system, while the is_Suspicious column reflects transactions deemed suspicious by compliance personnel. Figure 6 illustrates the attributes of the dataset along with their respective data types. The dataset is partitioned into training and testing sets, with a distribution ratio of 70:30. The heatmap and riskmlevel graph of dataset is presented in the figure 7 and 8. Is_suspecious and is_alerted attributers have lesser in a dataset. The risk level of is_suspecious and is_alerted are modetare values.

```
1 df_features.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1424 entries, 0 to 1423
Data columns (total 10 columns):
is_Alerted                1424 non-null int64
is_Suspicious             1424 non-null int64
transaction_amount        1424 non-null float64
correspondent_bank        1424 non-null object
debit_credit              1424 non-null object
Account_type              1424 non-null object
Account_Classification    1424 non-null object
Risk_level                1424 non-null object
Annual_income             1424 non-null float64
is_noncitizen             1424 non-null int64
dtypes: float64(2), int64(3), object(5)
memory usage: 111.3+ KB
```

Fig. 6. Data types of dataset attributes

The data ingestion is discussed in section 3.1 where it periodically reads the data from multiple sources and stores the data into data warehouse system. Once the data stored in the system, pre-processing steps starts at scheduled time. The data in data warehouse system is raw data that contains duplicate data, invalid data, outliers in data, and null values. So it is essential to clean and curate the data before it is feed to into LSTM model. The section 3.2 discusses how the data is preprocessed to remove above said inconsistent data. First step, outliers values are checked, if any outliers found then it are normalized to required range of values. Second step, duplicate check, where it validate the change degree of data, if it is within valid range, then data is used for further steps otherwise it will be deleted, third step is to null check, it is straightforward and delete the entire row if all the values are null. If one or two values are null then it will replaced with mean, or median values of column; further same procedure is followed for missing values also.

4.3 Hyperparameters

In the process of optimizing the Long Short-Term Memory (LSTM) model, nine hyperparameters are carefully tuned to enhance its performance. These hyperparameters include:

1. **Sliding Window Size (hp1)**: This parameter determines the length of the input sequences used for training the model.
2. **LSTM Units (hp2)**: Refers to the number of units or neurons in each LSTM layer, which affects the model's ability to capture patterns.

3. **Dropout Rate (hp3)**: This rate controls the proportion of neurons randomly dropped during training to prevent overfitting.
4. **Regularization (hp4)**: This includes techniques applied to avoid overfitting by penalizing large weights.
5. **Regularizer Rate (hp5)**: Specifies the strength of the regularization applied to the model.
6. **Optimizers (hp6)**: The algorithm used to update the model weights during training, which influences convergence and performance.
7. **Epochs (hp7)**: The number of complete passes through the training dataset, affecting the model's training duration and accuracy.
8. **Activation Function (hp8)**: The function used to introduce non-linearity in the model, which impacts how the network learns and generalizes.
9. **Learning Rate (hp9)**: The rate at which the model's weights are updated, crucial for achieving convergence and stable training.

Among these hyperparameters, the learning rate, number of epochs, and dropout rate are particularly critical. The learning rate influences how quickly the model adapts to the data, the number of epochs determines how thoroughly the model is trained, and the dropout rate helps in mitigating overfitting, thus directly impacting the overall accuracy and performance of the model.

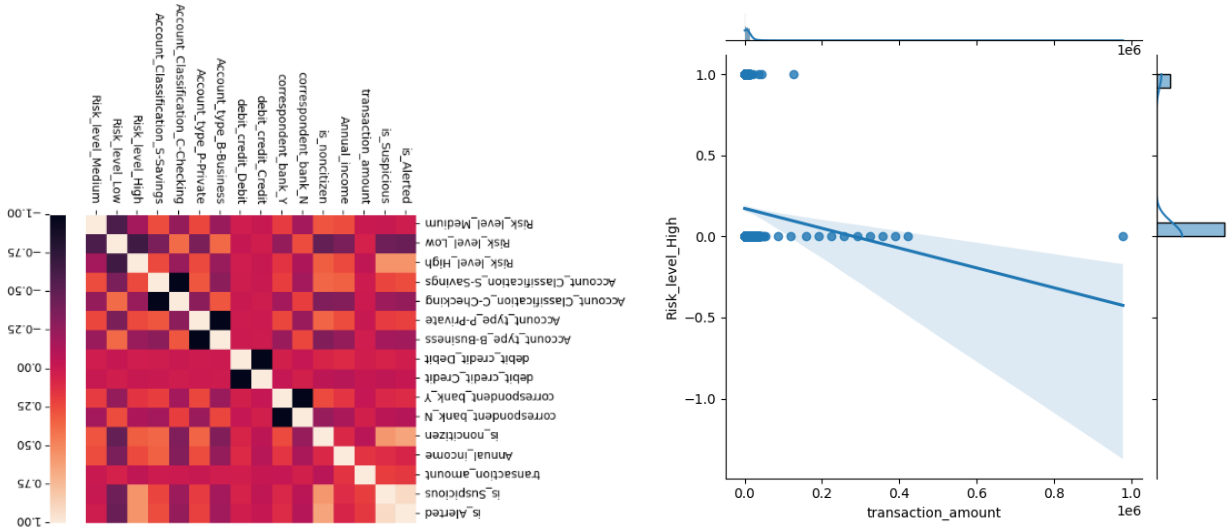

Fig. 7. Heap map and risk level graph of the transaction dataset

The risk level associated with transactions is determined by analyzing historical data in relation to the transaction amount. Figure 7 illustrates how the risk level varies with different transaction amounts. It has been observed that the risk level tends to increase for transactions within a specific range. Specifically, the risk is elevated for transactions of moderate amounts, while it is lower for both very small and very large transaction amounts. This suggests that transactions of moderate value pose a higher risk compared to those at the extremes of the amount spectrum.

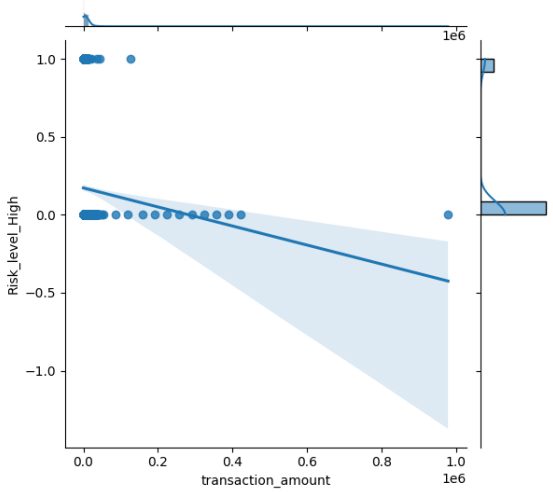

Fig. 8. Risk level for different transaction amount

4.4 The performance metrics:

Precision is a measure that tells us the percentage of correctly identified values. Simply put, it is a way to determine how accurately suspicious activity In order to evaluate the proposed model, we use commonly used performance metrics such as accuracy, recall, and the F1-score. It is important to consider these metrics when assessing the model's ability to detect suspicious behavior. In order to accurately calculate these performance measures, it is crucial to have a clear understanding of sensitivity and specificity. These two statistical indicators play a significant role in binary classification. Sensitivity, also known as the true positive rate, shows how well the model can accurately identify positive instances. On the other hand, specificity, also known as the true negative rate, measures how well the model can correctly identify negative instances. These metrics, when considered together, give us a complete understanding of how well the model is able to detect suspicious activities and how accurate it is in doing so.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{14}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{15}$$

$$F1\ Score = \frac{2*(Precision* Recall)}{Precision + Recall} \tag{16}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

$$Error\ Rate = \frac{FN+FP}{TP+FP+TN+FN} \tag{18}$$

There are several loss functions available for machine learning models, such as Mean Absolute Error Loss (MAE), Root Mean Squared Error (RMSE), and Mean Squared Logarithmic Error Loss (MSLEL). When you are evaluating a model for detecting suspicious activity in a time series, the best option to use is the Root Mean Square Error (RMSE). It is described as

$$Root\ Mean\ Squared\ Error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y^i - y_p^i)^2} \tag{19}$$

4.5 Experiment procedure:

We have implemented proposed model using Python programming language with predefined the libraries for various steps. The Python code is executed in Google Colab with different train and test dataset to capture the dynamism of proposed model. The implemented simulation model is configured with different epochs to validate the accuracy over different epochs. The ABC algorithm integrated into python code where it dynamically assigns values to hyperparameters and records the accuracy. The optimization algorithm changes values of parameters until maximum accuracy is recorded. Once the accuracy reaches to peak point or same accuracy values are recorded overs multiple iterations then algorithm stops its execution.

Results:

The initial result with default hyperparameters is an accuracy of 96.60% and a loss rate of 0.4% achieved in 117 seconds. When the bio-inspired algorithm is applied, the performance of the LSTM model improves. The LSTM model achieved a highest accuracy of 99.89% with a loss of 0.11. The proposed model needs more training, and whenever new data appears, it needs to be trained multiple times. This can consume more resources in order to achieve good accuracy. Table 2 displays sample simulation results that demonstrate the improved performance of the proposed model.

Table 2. Sample experiment results for tuning hyperparameters

Sl. no	sliding-window-size	LSTM-units	dropout-rate	regularize	regularizer-rate	optimizer	epochs	activation function	learning-rate	accuracy	Loss
1	10	30	0.4	L2	0.014	RMSProp	100	ReLu	0.6	96.6	0.4
2	20	30	0.3	L1	0.01	Adagrad	50	Sigmoid	0.5	96.9	0.32
3	30	40	0.25	L2	0.018	Adam	100	Tanh	0.4	98.2	0.28
..	..	..	..	..	.	…	..	..	..	..	..
n	20	30	0.25	L2	0..12	Adam	100	Sigmoid	0.25	99.89	0.11

In order to evaluate the suspicious activity detection model, the RMSE loss function is computed for both the test dataset and validation. Figure 9 shows the difference between the test and train loss for different epochs. The difference in loss between the test and train datasets is very small, and as the number of epochs increases, the loss continues to decrease. The accuracy of the suggested model for detecting suspicious activity is shown to be towards the higher end of the graph.

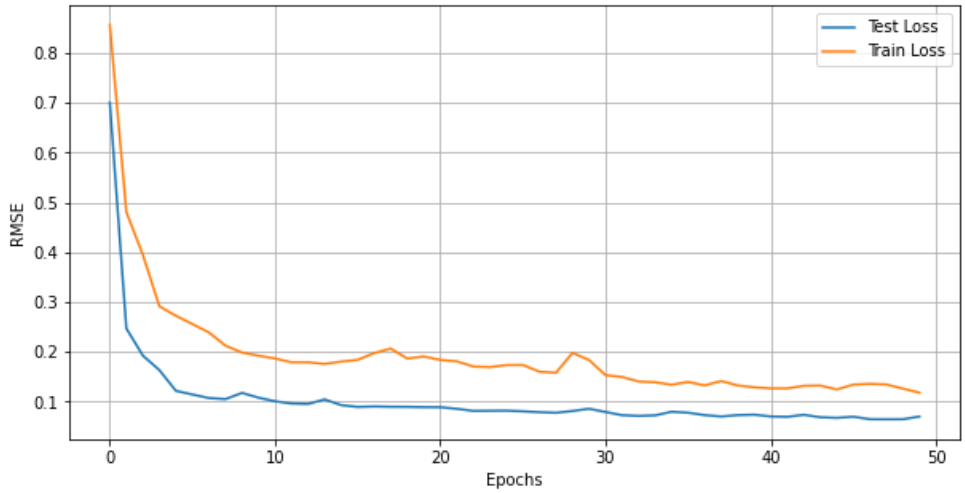

Fig. 9. RMSE for epochs

Figure 10 shows a comparison of the root mean square error (RMSE) for different models used in suspicious activity detection, such as HMM [21], DBN[32], LSTM [6] and ARIMA [27] A lower root mean square error (RMSE) indicates higher accuracy. The suggested model has a lower Root Mean Square Error (RMSE) when compared to other models for detecting suspicious behavior.

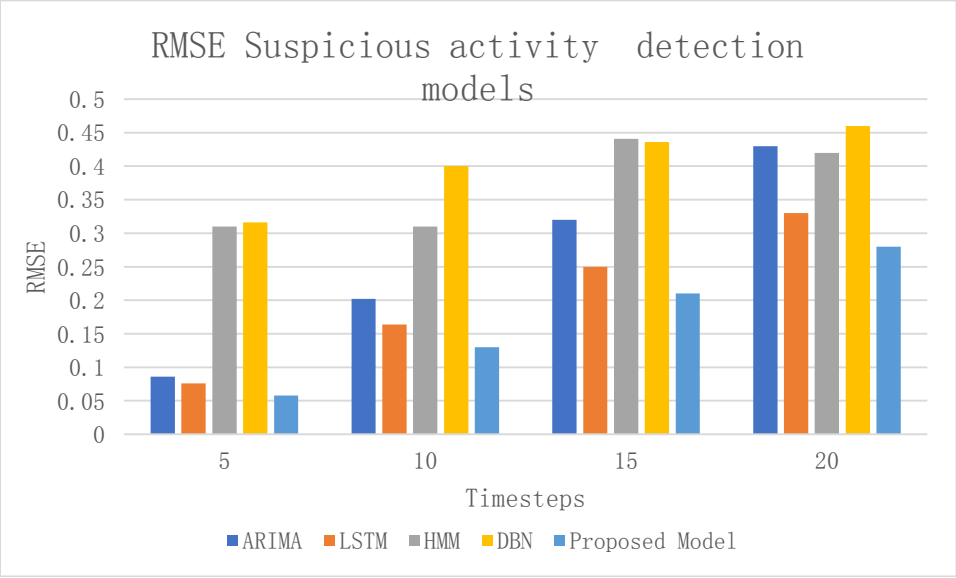

Fig. 10. Comparison of RMSE values for various models and proposed models.

The accuracy of the proposed model has increased compared to existing models. However, tuning the hyperparameters requires more memory and CPU cycles. So, the suggested method can be applied in critical situations where accuracy is extremely important. Figure 11 presents a comparison of various performance metrics, such as CPU and memory usage, detection time, and accuracy of suspicious behavior.

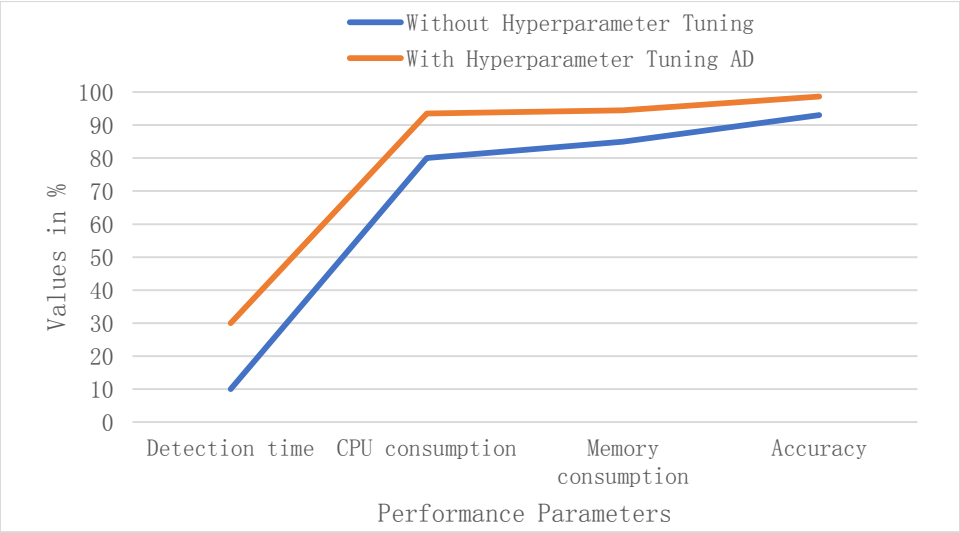

Fig. 11. Comparison of Normal AD and Hyperparameter Tuner AD

Figure 12 shows a comparison of the suggested method and other methods for finding suspicious behavior based on memory, accuracy, and F1-score. LSTMs [6], RNNs [28], and auto-encoders [26] are what the latest models are based on. Using the suggested model instead of the first three methods gives better precision, memory, and F1-score.

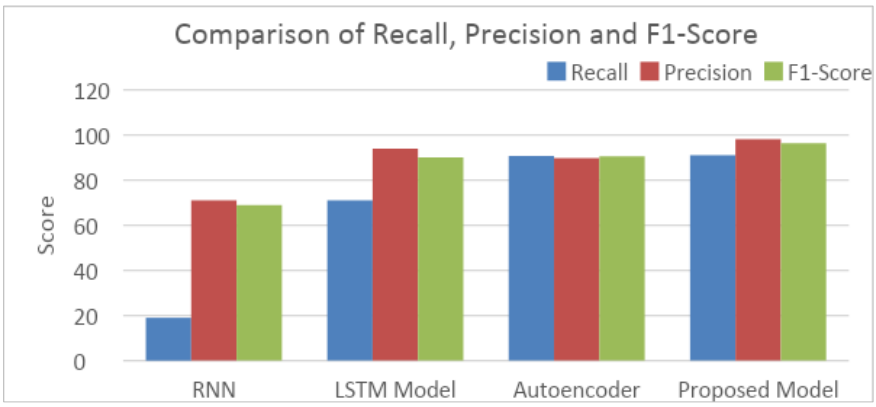

Fig. 12. Accuracy parameters of the existing and proposed system

Table 3 shows the set of hyperparameters that resulted in an accuracy rate of 99.89% using a threshold value of 0..65.The results of the experiments indicate that hyperparameter optimization outperformed models in terms of accuracy.

Table 3. Hyperparameter values for highest accuracy

Sliding-window-size	LSTM-units	dropout-rate	regularize	regularizer-rate	optimizer	epochs	activation function	learning-rate	accuracy	Loss
20	50	0.3	L2	0.02	RMSProp	100	ReLu	0.45	99.89	0.01

Table 4. Hyperparameter values for highest accuracy

References	Accuracy	Precision	F1-Score	Detection Time	CPU consumption	Memory Consumption	Scalability support
LSTM Autoencoder[26]	97%	92%	96.6%	More	More	More	Yes
LSTM [6]	92%	90	94	Less	Less	Less	NA
Auto-encoder [32]	95%	90%	95%	More	Moderate	Moderate	Yes
RNN [28]	89%	90%	90%	Moderate	Moderate	Moderate	No
HMM[21]	70	73.2%	72%	Moderate	Moderate	Moderate	No
DBN [32]	80.5714%	94.5440%,	95.3%	More	Moderate	Moderate	No
Proposed Model	99.8%	98.2%	98%	More	More	More	Yes

Table 4 shows comparison of proposed model and existing models with respect to accuracy, recall, precision, F1-Score, scalability, detection time, CPU and Memory Consumption. The data is collected from various research articles as part of literature survey. The proposed model recorded highest accuracy and support scalability as it is using comnination of Deep Learning Model and Bio-inspired algorithms in fog computing infrastructure.

Discussion

The banking and financial institutes are more concern about accurate data and results. They can invest more on accurate data. So this work, a data driven suspicious activity detection model is proposed to improve the accuracy and scalability and reduce the latency. The proposed suspicious activity detection model uses two popular technologies deep learning model and bio-inspired algorithm model. This combination is high effective and but complementary strengths of each method in managing complex data patterns and optimization tasks. The experiment results show the proposed model gains the high accuracy 99.89 as compared to existing state of art suspicious detection model.

The latency is very important parameter for the time critical application such as finance and medical application. The adoption of fog computing can reduce the latency as compared to cloud based applications. Most of popular banks are migrating to cloud based applications from on-prem services due to maintenance cost. Cloud based applications having more latency if cloud servers. Data centers to far away from customers and end users [34]. So, the replacement of fog computing infrastructure model with cloud computing infrastructure will reduce the latency. Further, fog computing infrastructure are having predefined configuration for scalability of applications. Fog computing model have load balancing and fault tolerance when uncertainties are happened in fog models; the fog nodes switch their task in between to balance the load and mitigate the issues. Hence the proposed model support scalability, minimal latency and provide high accuracy.

The major limitation of the proposed model is to higher computation resource consumption; figure 10 show the resource consumption for proposed model with hyperparameter optimization and without hyperparameter optimization; the proposed model consume more computation resource as compared to classical LSTM model.

5. Conclusion

One of the fundamental focuses to minimize financial loss for both bank segments and its customers is suspicious activity detection in your banking transaction. The core of this paper introduces a data driven approach to detecting suspicious activity in the bank customer transactions. A for infrastructure is employed to decrease the communication latency An LSTM model is used as a deep learning that will find the suspicious behaviour through some historical patterns of abnormal activities The LSTMs are subjected after initial training, using another bio-inspired optimizer called Artificial Bee Colony (ABC) algorithm hyperparameters fine-tuning this lead our LSTM-based detection models end in getting better accuracy. We evaluate proposed model with NAB temperature and CPU utilization data in Google Drive (from Colab). The result of the experiment will show that they outperform state-of-the-arte models in terms of accuracy to detect dubious activities. Experimental results show that the performance of proposed model is better on accuracy than existing models for detecting suspicious activities.

References

[1] Santomero, A.M. (1999). Risk Management in Banking: Practice Reviewed and Questioned. In: Galai, D., Ruthenberg, D., Sarnat, M., Schreiber, B.Z. (eds) Risk Management and Regulation in Banking. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-5043-3_3

[2] Chetan M Bulla, Satish S Bhojannavar, Vishal M Danawade,"Cloud Computing: Research Activities and Challenges",Volume 2, Issue 5, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2013.

[3] Resul Das, Muhammad Muhammad Inuwa,"A review on fog computing: Issues, characteristics, challenges, and potential applications",vo. 10, Telematics and Informatics Reports,2023.https://doi.org/10.1016/j.teler.2023.100049.

[4] Shi, S., Tse, R., Luo, W. et al. Machine learning-driven credit risk: a systemic review. Neural Comput & Applic 34, 14327–14339 (2022). https://doi.org/10.1007/s00521-022-07472-2

[5] Leo, M., Sharma, S., & Maddulety, K. (2019, March 5). *Machine Learning in Banking Risk Management: A Literature Review*. Risks. https://doi.org/10.3390/risks7010029.

[6] Fang Dao, Yun Zeng, Jing Qian,"Fault diagnosis of hydro-turbine via the incorporation of bayesian algorithm optimized CNN-LSTM neural network",vol 290, Energy,Elsevier, 2024.

[7] Al‐Sawwa, Jamil, and Mohammad Almseidin. "A Spark-Based Artificial Bee Colony Algorithm for Unbalanced Large Data Classification." *Information*, vol. 530, no. 11, 8 Nov. 2022, https://doi.org/10.3390/info13110530..

[8] Albatul Albattah & Murad A. Rassam. (2022) A Correlation-Based Anomaly Detection Model for Wireless Body Area Networks Using Convolution Long Short-Term Memory Neural Network. Sensors 22:5, pages 1951.

[9] Bulla, Chetan M., and Mahantesh N. Birje. "A Multi-Agent-Based Data Collection and Aggregation Model for Fog-Enabled Cloud Monitoring." *IJCAC* vol.11, no.1 2021: pp.73-92. http://doi.org/10.4018/IJCAC.2021010104

[10] Naghib, A., Jafari Navimipour, N., Hosseinzadeh, M. et al. A comprehensive and systematic literature review on the big data management techniques in the internet of things. Wireless Netw 29, 1085–1144 (2023). https://doi.org/10.1007/s11276-022-03177-5

[11] Pioli, L., Dorneles, C.F., de Macedo, D.D.J. et al. An overview of data reduction solutions at the edge of IoT systems: a systematic mapping of the literature. Computing 104, 1867–1889 (2022). https://doi.org/10.1007/s00607-022-01073-6

[12] Hongjian Li, Liangjie Liu, Xiaolin Duan, Hengyu Li, Peng Zheng, Libo Tang,"Energy-efficient offloading based on hybrid bio-inspired algorithm for edge–cloud integrated computation",vol 42,Sustainable Computing: Informatics and Systems,, elsevier, 2024.
[13] Sebin Jose, P. Victer Paul,"A survey on identification of influential users in social media networks using bio inspired algorithms,", Procedia Computer Science,Volume 218,2023.
[14] Bulla, C., Birje, M.N. (2022). Anomaly Detection in Industrial IoT Applications Using Deep Learning Approach. In: Fernandes, S.L., Sharma, T.K. (eds) Artificial Intelligence in Industrial Applications. Learning and Analytics in Intelligent Systems, vol 25. Springer, Cham. https://doi.org/10.1007/978-3-030-85383-9_9
[15] Aysha Shabbir, Maryam Shabir, Abdul Rehman Javed, Chinmay Chakraborty, Muhammad Rizwan, Suspicious transaction detection in banking cyber–physical systems, Computers & Electrical Engineering, Volume 97,2022,Elsevier
[16] A. Mahajan, V. S. Baghel and R. Jayaraman, "Credit Card Fraud Detection using Logistic Regression with Imbalanced Dataset," *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2023, pp. 339-342.
[17] Michael J.A., Gordon S.L.Data Mining Technique for Marketing, Sales and Customer Support,John Wiley & Sons INC, New York (1997), p. 445
[18] Yi-Cheng Shih, Tian-Shyr Dai, Ying-Ping Chen, Yen-Wu Ti, Wun-Hao Wang, Yun Kuo, "Fund transfer fraud detection: Analyzing irregular transactions and customer relationships with self-attention and graph neural networks", Expert Systems with Applications,2024,https://doi.org/10.1016/j.eswa.2024.125211.
[19] Alghuried, A. A Model for Anomalies Detection in Internet of Things (IoT) Using Inverse Weight Clustering and Decision Tree., Technological University, Dublin, 2017.
[20] P. Zhang and Y. Liu, "Application of An Improved Artificial Bee Colony Algorithm," IOP Conf. Ser.: Earth Environ. Sci., vol. 634, no. 1, p. 012056, Feb. 2021.
[21] Samir, A., Pahl, C.: Detecting and Predicting Anomalies for Edge Cluster Environments using Hidden Markov Models (HMM). In: 2019 Fourth IC-FMEC. pp. 21–28. IEEE, Rome, Italy (2019)
[22] Randhawa K., Loo C.H.U.K., Member S.,Credit card fraud detection using AdaBoost and majority voting, IEEE Access, 6 (2018), pp. 14277-14284, 10.1109/ACCESS.2018.2806420
[23] Guanjun L., Zhenchuan L., Lutao Z., Shuo W.,Random forest for credit card fraud. IEEE Access (2018)
[24] Aditi, Raut. (2023). Suspicious Activity Detection Using Machine Learning. International Journal For Science Technology And Engineering, 11(5):3745-3748. doi: 10.22214/ijraset.2023.52486.
[25] Wu, J., Yao, L., Liu, B., Ding, Z., Zhang, L.: Combining OC-SVMs With LSTM for Detecting Anomalies in Telemetry Data With Irregular Intervals. IEEE Access. 8, 106648–106659 (2020).
[26] Bulla, C., Birje, M.N. Improved data-driven root cause analysis in fog computing environment. J Reliable Intell Environ 8, 359–377 (2022).
[27] Demir, U., Ergen, S.C. ARIMA-based time variation model for beneath the chassis UWB channel. J Wireless Com Network 2016, 178 (2016). https://doi.org/10.1186/s13638-016-0676-3
[28] I. Ullah and Q. H. Mahmoud, "Design and Development of RNN Anomaly Detection Model for IoT Networks," in IEEE Access, vol. 10, pp. 62722-62750, 2022, doi: 10.1109/ACCESS.2022.3176317. keywords: {Internet of Things;Security;Deep learning;Intrusion detection;Computational modeling;Recurrent neural networks;Telecommunication traffic;Internet of Things;anomaly detection;recurrent neural network;convolutional neural network;LSTM;BiLSTM;GRU},
[29] Maya, S., Ueno, K., Nishikawa, T.: dLSTM: a new approach for anomaly detection using deep learning with delayed prediction. Int.Jou.of Dat.Sciee Ana. 8, 137–164 (2019).
[30] B. V. V, V. Indhuja, M. V. Reddy, N. Nikhitha and P. Pramila, "Suspicious Activity Detection using LRCN," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 1463-1470, doi: 10.1109/ICSSIT55814.2023.10061045.
[31] Kamal, Upreti., Prashant, Vats., Aravindan, Srinivasan., K., V., Daya, Sagar., R, Mahaveerakannan., G., Charles, Babu. (2023). Detection of Banking Financial Frauds Using Hyper-Parameter Tuning of DL in Cloud Computing Environment. International Journal of Cooperative Information Systems, doi: 10.1142/s0218843023500247
[32] Talapula, D.K., Kumar, A., Ravulakollu, K.K., Kumar, M. (2023). Anomaly Detection in Online Data Streams Using Deep Belief Neural Networks. In: Swaroop, A., Kansal, V., Fortino, G., Hassanien, A.E. (eds) Proceedings of Fourth Doctoral Symposium on Computational Intelligence. DoSCI 2023. Lecture Notes in Networks and Systems, vol 726. Springer, Singapore. https://doi.org/10.1007/978-981-99-3716-5_59
[33] Torabi, H., Mirtaheri, S.L. & Greco, S. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity* **6**, 1 (2023). https://doi.org/10.1186/s42400-022-00134-9
[34] Meenakshi, Ravindra S, Girish Wali, Chetan Bulla, Jitender Tanwar, Madhava Rao Chunduru, Surjeet, "AI Integrated Approach for Enhancing Linguistic Natural Language Processing (NLP) Models for Multilingual Sentiment Analysis",Vol. 23 issueNo. 1, Linguistic and Philosophical Investigations, (2024).
[35] Wali, G., Kori, A., Bulla, C., & AIML, K. Market Risk Assessment Using Deep Learning Model and Fog Computing Infrastructure.

Authors' Profiles

Girish Wali completed his masters in Business Administration and currently working as SVP - Business Architecture, IT Business Lead Analyst, Business Analysis & Product Solutions at Citi. The area of interest is Deep Learning model Gen AI in Business Model.

Chetan Bulla, PhD, he is a Professor of Computer Science and Engineering. He received his Ph.D. and M.Sc. degrees in Cloud Technology. He is the author of more than 30 scientific contributions including articles in international Peer-review Journals and Conferences in the area of Wireless Mobile communications. His area of interest cloud computing, fog computing and AI.

How to cite this paper: Girish Wali, Chetan Bulla, "Suspicious Activity Detection Model in Bank Transactions Using Deep Learning with Fog Computing Infrastructure", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.16, No.6, pp. 1-17, 2024. DOI:10.5815/ijieeb.2024.06.01