iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68356

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Online Fraud Detection Using Hidden Markov Model and Behavior Analysis

Faith R Kakwere[1], Zheng Taotao[2]
School of Science, Zhejiang University of Science and Technology, Hangzhou, China

Abstract: ***In the realm of online shopping, credit cards have emerged as the preferred method of payment for many consumers. However, with their widespread use comes a troubling rise in fraudulent activities. The nature of online transactions makes them particularly vulnerable, as they require only the card details, which can be easily stored in digital formats, rather than the physical card itself. Unfortunately, current systems typically identify fraudulent transactions only after they have been completed, leaving consumers at risk.***

This paper introduces a promising solution utilizing Hidden Markov Models (HMM), a sophisticated stochastic approach designed to analyze systems that exhibit variability. By establishing specific threshold values, this innovative system can effectively discern between legitimate and fraudulent transactions, boasting an impressive accuracy rate of nearly 80% and rapid processing capabilities, as supported by recent studies.

Moreover, Behavioral Analysis (BA) serves an important aspect in deciphering the spending patterns of cardholders, enhancing the system's ability to detect fraud. The integration of Hidden Markov Models allows for a cost-effective strategy to combat fraud, leveraging the forward-backward algorithm to achieve remarkable results. Together, these advancements pave the way for a safer online shopping experience, empowering consumers and fostering trust in digital transactions.

Keywords: Fraud Detection System (FDS), Card Holder, Transaction, Fraud Detection, Hidden Markov Model (HMM), Behavior Analysis (BA), Hidden Markov Model forward and backward algorithms.

I. INTRODUCTION

As the digital economy advances quickly, companies of every size must reassess their stance and resources regarding fraud management.Transaction fraud poses real threats to online shopping.As the popularity of online transactions grows, the variety of frauds associated with them is also rising, causing disruptions in the banking sector. Digital remittances are a main aim at fraudsters since they do not need the actual card, just the card information that can be stored electronically.

To address these issues, many fraudulent unmasking methods and algorithms suggested and data mining is utilized through various companies involved in fraud detection. However, data mining by itself is inadequate for identifying fraud because it relies on data set that includes the history of customer transactions. More-so transaction patterns show variations and agent trends with increased spending trends during school seasons and major holidays without ignoring the current economic situations. The details of things someone buys in each transaction are usually unknown to any Fraud Detection System (FDS) used by the bank that gives out credit cards. Behavior Analysis (BA), are used to address this issue. Every incoming transaction is submitted to the FDS for verification. FDS collects the card information and transaction fee to verify the legitimacy of the transaction[1,2], and the bank declines the transaction if it is determined to be fraudulent by FDS.

The primary goal of this study is to create and assess a Hidden Markov model which analyzes card holder's buying behaviors across merchant categories and accurately allocate a distinct pattern based on prevalent transaction trends.This study uses the forward and backward algorithms of the HMM to analyze, refine, and predict buying behaviors of card holders engaging specific merchant categories which illustrate the worldwide card payment landscape.

II. SUPPORT VECTOR MACHINES

Support vector machines have been applied to predict the card holder's buying patterns. A notable example is Shobana, J., et al.[3] who utilized support vector machines to forecast e-commerce, card and customer purchasing behaviors preventing churn. Researchers trained support vector machines using labeled datasets, enabling them to accurately predict and recognize outcomes.

©IJRASET: All Rights are Reserved

To estimate customer attrition in transaction-to-consumer e-commerce, researchers implemented a dual structure system. Initially the method used support vector machine technology to anticipate churn events, on the other hand the second method incorporated an approach combining cooperative, substance-oriented, information-driven, and demographic techniques to formulate tailored reserved strategies. Researchers determined that calculating the worth of strayed customers suggested that with the number and frequency of transactions increased, the likelihood of consumer attrition significantly decreased. Although the researchers obtained satisfactory results with a prediction accuracy of 77.36% by coaching the support vector machines, it was recognized that extensive coaching time needed by support vector machines, particularly in dealing with huge data sets.

A. K-nearest neighbor

John Awoyemi[4] conducted a search utilizing Naive Bayes, K-nearest neighbor and logistic regression. They completed their tasks in Python. These were used on transaction data and subsequently resampled using the Synthetic Minority Over-Sampling Technique. According to the findings, K-nearest exhibited the highest performance of the three, evaluated based on receptiveness, particularity, accuracy, interdependence, and relevance.

B. Bayesian classifier

Similarly, Edoardo, R.,et al.[5] characterized a Bayesian classifier as a statistical method for calculating the likelihood that a feature is part of a class using Bayes' theorem. Researchers noted that Naive Bayes operates under the premise of independence among predictors, is capable of managing huge data sets, which makes it straightforward to construct with the two components constituting this algorithm: Naïve and Bayes. Recently, scholars investigated the potential of Naïve Bayes, particularly when predicting credit card and customer behaviors.

C. Machine Learning

Of late Machine Learning (ML) has become popular due to its accuracy hence most fraud detection industries are transitioning from Traditional Fraud Detection to Machine Learning Fraud Detection. A few distinctions between Machine Learning Fraud Detection and Traditional Fraud Detection are ,for Rule based methods uses traditional methods which are programmed with manual algorithms that allow them to detect the most obvious fraud and multiple authentication methods are always used which can cause problems for users. On the other hand ML can process data in real time and the algorithms work with BA to help reduce verification processes.

D. Hidden Markov Model

The Hidden Markov Model is grounded on intensifying the Markov Chain. A Markov Chain is a structure that illustrates the probability of a sequences of states and random variables. Every state occupies certain values from various sets. Hidden Markov Models have been effectively utilized in various fields including speech recognition, traffic congestion forecasting, robotics, bio informatics, and data mining. In a Markov chain, the theory is robust for predicting the future, relying solely on the present state. This essentially indicates that the state preceding the current state will not affect the future state; only the current state is significant (Markov Property). However, in certain situations where we aim to compute probabilities, these states are concealed and cannot be observed. For instance, we cannot see a transition occurring. Consequently, for those occurrences, the Hidden Markov Model (HMM) is utilized, enabling the determination of the two probabilities, visible and concealed events.

A Hidden Markov Chain consists of the following components Q = $\{q_1, q_2, \ldots q_N\}$,are a set of N states,

$A = \{a_{11}, a_{12}, \ldots, a_{n1}, \ldots, a_{nm}\}$,are Transition probability matrix A, where each a_{ij} states the probability of transitioning from state to i to j .

$\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$, are initial probability distribution over states.

$O = \{o_1, o_2, \ldots, o_N\}$, are sequence of Mobservations

$B = b_i(o_t)$, are likelihood observations, with the probability of observation o_t originating at state i .

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

III. TECHNIQUES AND ALGORITHMS

As mentioned in the previous section, to fully specify the HMM, we require a pair of estimated structural parameters, N and M , and A, B and π as the combined distributions.Which are rewritten as $\hat{\lambda} = \{A, b, \pi\}$.

Typically the forward-backward, also known as the Baum-Welch algorithm is used to train HMM. The algorithm will allow us to train both the transition probabilities A of the EM and the emission probabilities B of the HMM. EM is an iterative process that starts with an initial estimate for the probabilities, then refines these estimates to obtain better values, continuously enhancing the probabilities it learns through repetition.

A tutorial by[6] , presented the concept that hidden Markov models must be defined by three core issues :

Likelihood: Given an HMM λ = (A,B) and an observation sequence O, determine the likelihood P(O|λ) which is $P(O \mid Q, \hat{\lambda}) = \pi_{t+1}^{N}(O_t \mid q_t, \hat{\lambda})$.

The likelihood probability written as $P(O \mid Q, \hat{\lambda}) = b_{q_1}(O_1) * b_{q_2}(O_2) \cdots b_{q_N}(O_N)$ Decryption: Given an observation sequence O and an HMM λ = (A,B), discover the best hidden state sequence Q, $P(Q \mid \hat{\lambda}) = \pi_{q_1} a_{q_1 q_2} * a_{q_2 q_3} \ldots a_{q_{N-1} q_N}$.

Training: Provided sequence O as the observation and the set of states in the HMM, train the HMM parameters A and B. Hence by adding all possible sequence states initially , we calculate a combined probability of being in a certain state Q and producing O, generally this is $P(Q \mid \hat{\lambda}) = \sum_{\forall Q} P(O \mid Q, \hat{\lambda}) P(Q \mid \hat{\lambda})$ in [7,8]. However, for real life situations in which the number of hidden observations N, and sequences T, N^T becomes very large and cannot be computed separately hence we use the forward algorithm.

A. Forward Algorithm

The Forward Algorithm is defined as a method used in Hidden Markov Models to compute the probability of a sequence of observed outputs up to a certain time, given the current state and model parameters. It efficiently calculates the forward probabilities for all intermediate states using a state web.

Every cell in the forward algorithm trellis $\alpha_t(j)$ signifies the likelihood of being in state j after observing the initial t observations, given the automaton λ. The value of each cell$\alpha_t(j)$is calculated by adding the probabilities of all paths that might lead us to this cell. The forward algorithm, in which forward[s, t] imitates $\alpha_t(s)$has 3 stages which are :

1. Initialization:$\alpha_1(j) = \pi_j b_j(o_1) \, 1 \le j \le N$.

Recursion: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \;\; 1 \le j \le N, 1 < t \le T$.

3. Termination: $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

B. Viterbi algorithm

Viterbi algorithm for determining the best sequence of hidden states. Provided an observational sequence and an HMM λ = (A, B), the algorithm yields the state path within the HMM that allocates the highest likelihood to the observation sequence.The mostcrucial thingis the fact that Viterbi algorithm is the same as the forward algorithm, with the difference being that it uses the maximum of the prior path probabilities, while the forward algorithm uses their sum. Additionally, it's important to mention that the Viterbi algorithm includes a feature that the forward algorithm lacks: back-pointers. The reason is that the forward algorithm needs to generate an observation likelihood, whereas the Viterbi algorithm must provide a probability and additionally the most probable state sequence. We determine the optimal state sequence by monitoring the progression of hidden states that resulted in each state and then subsequently tracing back the best route to the start (the Viterbi back-trace).

In conclusion, the precise interpretation of the Viterbi recursion as stated below :

1) Initialization: $v_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N \; ; \;\; bt_1 = 0, \;\; 1 \le j \le N$

2) Recursion:

©IJRASET: All Rights are Reserved

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij} b_j(o_t);\ 1 \le j \le N, 1 < t \le T;$$

$$bt_t(j) = \operatorname{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij} b_j(o_t); 1 \le j \le N, 1 < t \le T.$$

3) Termination:

The best score: $P^* = \max_{i=1}^{N} v_T(i)$

The start of backtrace: $q_{T^*} = \operatorname{argmax}_{i=1}^{N} v_T(i)$

E. The forward-backward algorithm

The forward–backward algorithm calculates the probabilities of every potential state sequence that might contribute to the generation of the target observation sequence. The conventional method for training HMMs is the forward-backward, or Baum-Welch algorithm [9], which is a specific instance of the Expectation-Maximization (EM) Baum-Welch algorithm [10]. The algorithm is going to make it possible to train A which is the transition probability and B, the emission probability of the HMM using Expectation Maximization. EM is a repeated equation that generates initial estimates for the probability, subsequently making use of the estimates to derive a more accurate estimate, and continues this process, progressively enhancing the probabilities it learns. However, the true issue is even more challenging: we lack knowledge of the numbers when residing in any of the concealed states! The Baum-Welch equation/algorithm addresses this when it repeatedly estimates the counts. We will begin with an estimation of the transition and observation probabilities and subsequently utilize these estimated probabilities to obtain increasingly accurate probabilities. We will achieve this by calculating the forward probability for an observation and subsequently distributing that likelihood combination between the various pathways that helped create this forward probability.

Towards grasping the equation, we must establish the relevant possibility linking the forward probability, known as the backward probability. This is calculated inductively in a way analogous to the forward algorithm.The forward and backward probabilities assist in calculating the transition probability a_{ij} and observation probability $b_i(o_t)$ from an observation sequence, despite the fact that the true path through the model remains concealed. First we observe how to estimate $\hat{a}_{ij}$ by a variant of simple maximum likelihood estimation:

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}.$$

Let's suppose we possessed an estimate regarding the likelihood that a specific transition $i \to j$ occurred at a certain moment t within the observation sequence. Understanding this likelihood for every specific time t, can be aggregated on all times $t+1$ approximating the total number of transitioning $(\xi)_t \quad i \to j$, expressed as :

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda).$$

To calculate ξ_t ,we initially determine a probability that resembles ξ_t, but differs by incorporating the likelihood of the observation; paying attention to the varied condition of O, the α and β probabilities, the transition probability a_{ij} and the observation probability $b_j(o_{t+1})$ were combined to produce : not-quite $\xi_t(ij) = P(q_t = i, q_{t+1} = j | O, \lambda)$.

Hence not-quite $\xi_t(i,j) = \alpha_t a_i jb_j(o_{t+1}) \beta_{t+1}(j)$.

To calculate ξ_t given not-quite ξ_t , trail the law of probability divided by $P(O|\lambda)$, knowing well $P(X|Y,Z) = \frac{P(X,Y|Z)}{P(Y|Z)}$.

The likelihood of the observations based on the structure is essentially the forward probability of the entire expression.

Therefore, the ultimate equation for ξ_t is $\xi_t(ij) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\, \beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j)\, \beta_t(j)}$.

The anticipated count of transitions between i and j states is thus the aggregate of all of t of ξ. To compute our estimate of a_{ij}, we require one additional element : the overall anticipated count of transitions from state i. This is achieved through calculating the sum of all transitions from state. Finally

$$\hat{a}_{ij}: \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}. \quad (1)$$

We require a concept for recalculating the observation likelihood. Therefor representing the likelihood of a specific symbol v_k from the observation, at a specified state j : $\hat{b}_j(v_k)$. This is obtained by calculating the numerator of equation (2) is merely the result of the forward probability multiplied by the backward probability:

$$\gamma_t(j) = \frac{\alpha(j)\beta(j)}{P(O|\lambda)}. \quad (2)$$

We are prepared to calculate b. For the numerator, we add together $\gamma_t(j)$ all time steps t where the observation matches o_t is the symbol v_k we are focused on.For the denominator, we calculate $\gamma_t(j)$ the total step time across t. Therefor the outcome represents the ratio of occasions during state j and observed symbol v_k; "sum across all t for which the observation at time t was v_k "): $\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$.

We currently possess methods in equations (1) and (2) to recalculate the transition A and observation B probabilities based on an observation sequence O, given that we already have prior estimates of A and B. These recalculations are fundamental to the iterative forward-backward algorithm. The forward-backward algorithm begins with an initial estimation of the HMM parameters $\lambda = (A, B)$. We subsequently perform two steps repeatedly. Similar to other instances of the EM (expectation-maximization) algorithm, the forward-backward algorithm consists of two phases: the expectation phase, known as the E-step, and the maximization phase, referred to as the M-step. During the E-step, we calculate the anticipated state occupancy count γ and the anticipated state transition count ξ based on the previous A and B probabilities. M-step, utilizes γ and ξ for the recalculation of the probabilities A and B.

F. Detection Algorithm

We will obtain the HMM domains for the registered users. The forward-backward algorithm will start with all the initial parameters and will collect the closest values. This observation sequence will be formed based on the card holder's transaction history over time. This input sequence will be incorporated into the HMM to determine the likelihood of approval. Assuming we obtain probability α_1 as $\alpha_1 = P(o_1, o_2, \ldots o_N \mid \lambda)$.

Viewing O_{N+1} as a recent sequence for a specific moment $t+1$, during transacting is underway. Now that we have a sequence, to consider only $N+1$, to take into account only N sequence, O_1 has to be eliminated and thus sequences from O_2 to O_{N+1} will be accounted for.

We obtain the updated probability as $\alpha_2 = P(o_2, o_3, \ldots o_{N+1} \mid \hat{\lambda})$. Which results to : $\Delta\alpha = \alpha_1 - \alpha_2$

If we obtain results as $\Delta\alpha > 0$, HMM will consider a new progression O_{N+1} with the smallest probability, hence the process will be deemed fraudulent only if the change in the probability percentage exceeds the previously established thresholdvalue : $\Delta\alpha/\alpha_1 \geq threshold$.

Regarding this system, three price tiers are established for the expenditure profile of any cardholder: Large (L) starting at ($651 to the absolute amount), Median (M) commencing at ($251 to $650), and Small (S) covering ($0 to $250). So viewing the set of symbols as V = [S, M, L]. We will likewise specify HMM domain such as Matrix A for transition state probability, Matrix B for observation symbol probability, and Vector π for initial state probability. Entirely these three factors will be taken into account during the training phase of the HMM. There will be transactions for different items of different categories with different amounts. This will make use of deviation in purchasing amount of the last 10 transaction with addition of one new transaction, which is a possibility for getting probability.

During the initial stage, model is not fed with more transactions, therefore while registering, user will be asked to provide some security question along with answers. With the help of all these details, the HMM model obtains the information for prospective validation by analyzing each user's expenditure patterns.

Fraud Detection Phase Steps:

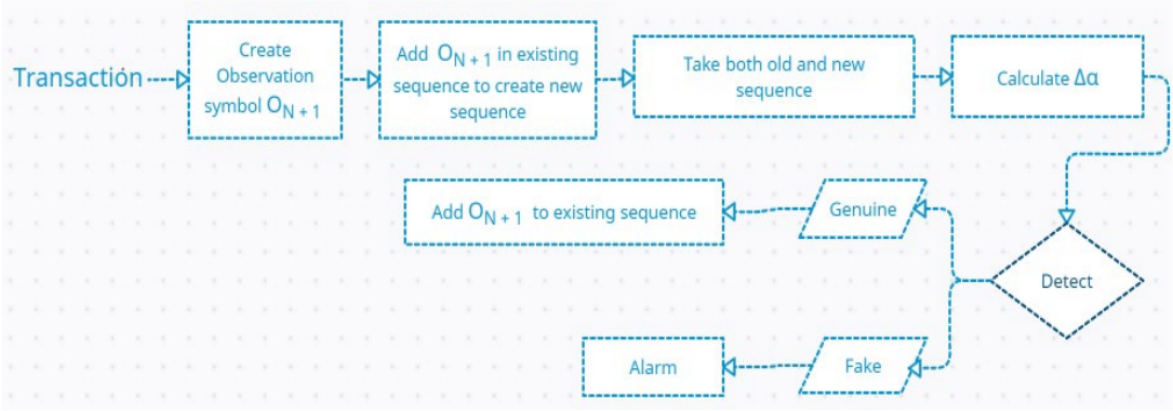


Figure 1 : Detection Algorithm Architecture

IV. FRAUD DETECTION

When the user initiates the payment, card information will be needed. This encompasses card information like the card verification value (cvv), time of expiration like the year and month. After the user inputs the data, it will be verified against the card database. Once these details are verified, if the system detects any discrepancy between the current transaction and previous transactions, the detection module will be triggered. As this framework is structured as any e-commerce site, the information of enrolled customers, purchase specifics, and transaction amounts will be kept in the back-end SQL database tables. If the user has fewer than 10 transactions in the database, the system will directly request all personal information for the transaction to proceed. Once the database contains over 10 transactions, it will begin comparing with prior transactions before commencing its operations.

The Hidden Markov Model will retrieve information stored in the SQL database tables which will be trained according to the user's spending patterns, with data being categorized based on a predefined threshold value. The amount and category of the current purchase will be compared with those of previous transactions. Upon completing the computation for transaction likelihood, it will determine whether the transaction is valid or fraudulent. If the system identifies the transaction as fraudulent, an additional verification step will be included. This is the form for the security question that the user established during registration.

If the user provides accurate responses, the transaction will proceed. If the business deal is fraudulent and incorrect solutions will be provided, thereafter three trials, the card and account will be disabled, hence deal will not be permitted.
If a user fails to remember the answers to the security questions, they can complete the unblock request form to ask for their account to be unblocked. Currently this unblock request is kept simple and does not include much security because currently we are mainly focusing on detecting fraud. Figure 2 illustrates an activity diagram of the suggested system.

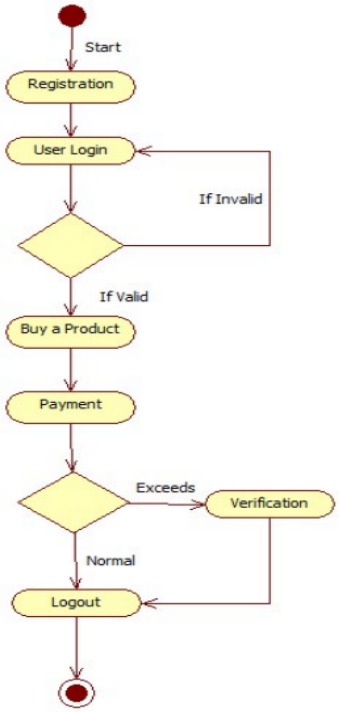


Figure 2 : Proposed system.

V. CONCLUSION

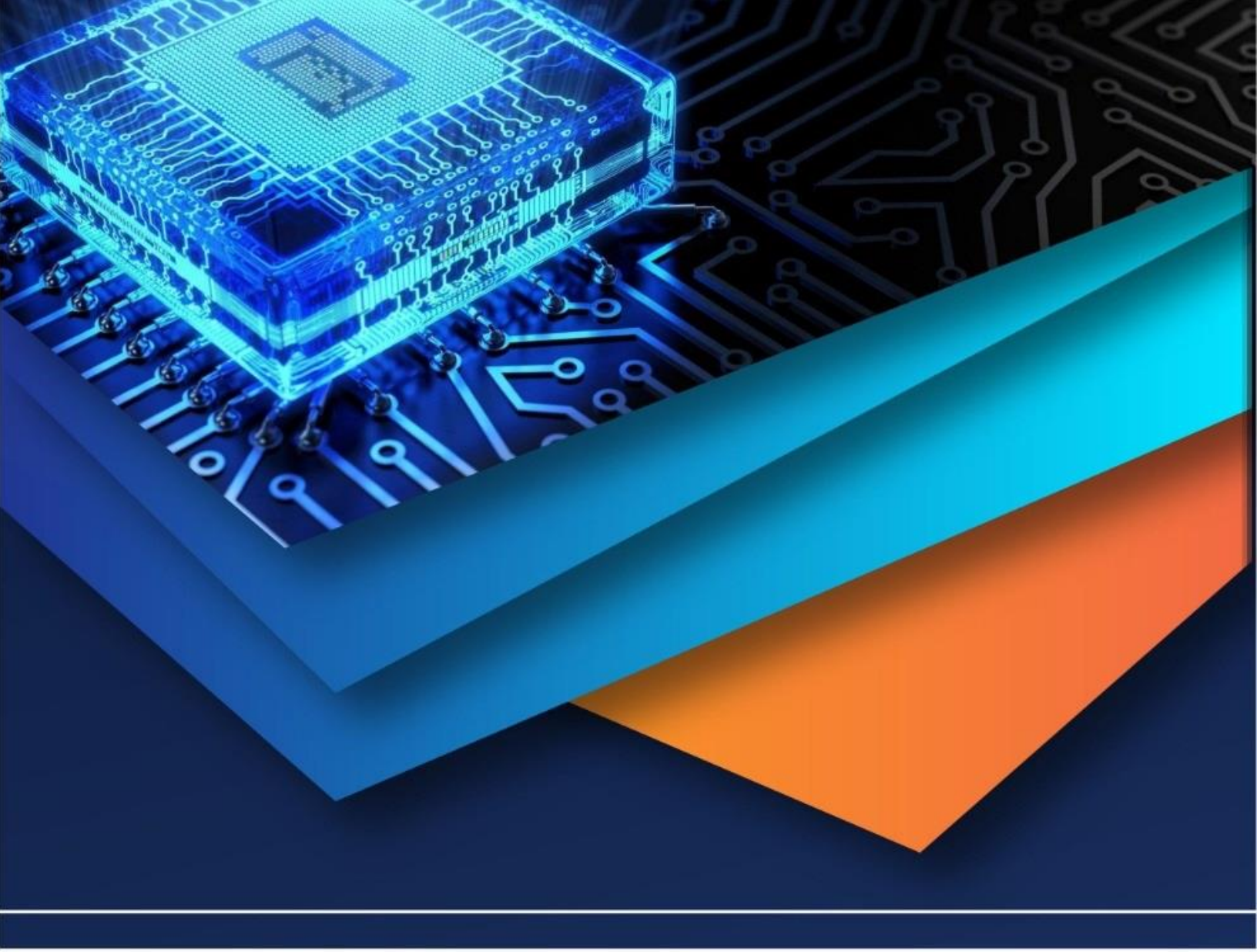
The suggested structure, explored the effectiveness of HMM Markov for identifying fraudulent activities in online transactions. Every stage involved in the business procedure is regarded as stochastic processes of Hidden Markov Models, whereas the price intervals of transactions are viewed as observation symbols, and the purchased items are recognized as states within the Hidden Markov Model (HMM). The suggested system is also capable of scaling to manage large volumes of transactions. This system provides rapid results compared to the current system. Within the structure, every new transaction is evaluated in terms of authenticity or fraud according to the consumer's expenditure patterns. This structure will additionally determine whether transactions are fake or legitimate according to the established threshold values. The Fraud Detection system achieves an accuracy of almost 80% and demonstrates a high processing speed, as indicated by the comparative studies conducted. It is highly appropriate for detecting Online Transaction Fraud since it keeps a record of users, eliminating the need to verify the original user.

REFERENCES

[1] Mhatre, G., Almeida, O., Mhatre, D., & Joshi, P. (2014). Credit card fraud detection using hidden markov model. International Journal of Computer Science & Information Technology, 5(1), 37--48.
[2] Ghosh, S., & D.L. Reilly, D. L. (1994). Credit Card Fraud Detection with a Neural-Network, Proc. 27th Hawaii Int"l Conf. System Sciences: Information Systems: Decision Support and Knowledge-Based Systems, 3, 621--630.
[3] Shobana, J., Gangadhar, C., Kumar, R., Renjith, P., Bamini, J., & Chincholkar, Y. (2023). E-commerce customer churn prevention using machine learning-based business intelligence strategy. Science Direct Articles on Measurement, 27, 1--8.
[4] John, O. A., Adebayo, O. A., & Samuel, A. O. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. International Conference on Computing Networking and Informatics (ICCNI), 1--9.
[5] Edoardo, R., Voroli, C., &Farcomeni, A. (2023). Quantile-distribution functions and their use for classification, with application to naïve Bayes classifiers. Springer Journal on Statistics and Computing, 33(55), 1--15.

[6] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE, 77(2),257--286.
[7] Klabunde, R., Daniel Jurafsky& James h. Martin. (2002).Speech and language processing. Zeitschrift für Sprachwissenschaft, 21(1), 134--135.
[8] Stamp, M. (2004). A revealing introduction to hidden Markov models. Department of Computer Science San Jose State University, pp.26--56.
[9] Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities III: Proceedings of the 3rd Symposium on Inequalities. Academic Press:1--8.
[10] Dempster, A. P., N. M. Laird, and D. B. Rubin. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39(1):1--21.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)