DeLClustE: Protecting Users from Credit-Card Fraud Transaction via the Deep-Learning Cluster Ensemble

Fidelis Obukohwo Aghware[1], Rume Elizabeth Yoro[2], Patrick Ogholoruwami Ejeh[3], Christopher Chukwufunaya Odiakaose[4], Frances Uche Emordi[5], Arnold Adimabua Ojugo[6]

Department of Computer Science-Faculty of Computing, University of Delta, (UNIDEL) Agbor, Delta State, Nigeria[1]
Department of Cybersecurity-Faculty of Information Tech., Dennis Osadebay University Asaba, Delta State, Nigeria[2, 5]
Department of Computer Science-Faculty of Information Technology, Dennis Osadebay University Asaba, Delta State, Nigeria[3, 4]
Department of Computer Science-College of Science, Federal University of Petroleum Resources Effurun, (FUPRE), Delta State, Nigeria[6]

***Abstract*—Fraud is the unlawful acquisition of valuable assets gained via intended misrepresentation. It is a crime committed by either an internal/external user, and associated with acts of theft, embezzlement, and larceny. The proliferation of credit cards to aid financial inclusiveness has its usefulness alongside it attracting malicious attacks for gains. Attempts to classify fraudulent credit card transactions have yielded formal taxonomies as these attacks seek to evade detection. We propose a deep learning ensemble via a profile hidden Markov model with a deep neural network, which is poised to effectively classify credit-card fraud with a high degree of accuracy, reduce errors, and timely fashion. The result shows the ensemble effectively classified benign transactions with a precision of 97 percent. Thus, we posit a new scheme that is more logical, intuitive, reusable, exhaustive, and robust in classifying such fraudulent transactions based on the attack source, cause(s), and attack time gap.**

Keywords—Fraud transactions; fraud detection; deep learning ensemble; credit card fraud; cluster modeling; financial inclusion

I. Introduction

The rise in the adoption of computing devices to aid effective data processing and resource sharing has continued to attract adversaries. This has necessitated the deployment of systems to avert such threats. The growth in these attacks has also resulted in higher costs associated with the safeguarding of valuable resources shared across networks [1]. Attackers have become more proficient at exploiting flaws with access to privileges, aimed at financial gains – even with advances made in the medium of data sharing [2]. This remarkable evidence advances a digital revolution such that day-to-day living is impacted therein with the proliferation of buying/selling via such mode, platform(s), and adoption of credit cards that have consequently, exposed many users to more clever and complicated methods to steal considerable money [3]–[5]. The growing complexity of ICT and the frequency of threats have also increased the data required to successfully detect them. There is also a rise in the adoption of multi-staged, subterfuge attacks targeted at various levels of security as provisioned in many organizations. Another barrier to detection is that adversaries often disguise the true forms and nature of their assault – and rarely, take up abrupt spurts of suspicious behavior that are easily recognized by simple intrusion detection schemes [6]–[9].

Previous studies have continued to acknowledge the rise in trend/alarming growth in credit card fraud, which has continued to lower user trust (irrespective of the rise in the adoption of credit cards) [10]. Studies also note that such fraudulent activities have caused greater losses to the financial services industry. This has thus, positioned as imperative – many researchers that adopted statistical models in detecting malicious credit card transactions [11]. Implementing a stochastic model has its bottleneck – as malicious transactions are aimed to evade detection, and their respective performance is often hindered by model over-fitting, parameter selection, etc [12].

The limited availability of data and 'censored' results from previous studies – have also led to difficulties to advance this field as datasets contain ambiguities, partial truth, and noise. These, have led to improper selection of features, data encoding, poor learning convergence, and incorrect results from over-parameterizing, overfitting, and overtraining. This increases false-positives and true-negatives error rates. We resolve this via a robust search that will effectively classify observations and yield the expected values [13]–[16].

The continued complexity in credit-card fraud detection has left us all in a frenzy with the continued quest to tweak methods to evade detection (for adversaries) as well as means to curb all attacks/threats (for security experts). This, in turn, has made and left such task and business, both a continuous and inconclusive feat [17]. In the quest therein for improved frameworks, some studies have shown that such tasks also, yield models whose performance is continually degraded at intervals due to improper selection of features within the used dataset for training and testing therein [18]–[20]. Even with the use and adoption of intelligent, stochastic, and dynamic classifiers, credit-card fraud persists as adversaries continue to evolve their techniques.

Thus, our study seeks to explore the use of feature selection [21]–[23] that is capable of addressing the issues of optimization with appropriate feature(s) selection, and adequately training the framework to avoid pitfalls from over-parameterization and overfitting of the model using deep learning. We propose a deep-learning cluster model to aid credit-card fraud detection. This will help to explore, exploit and use observed data as well as seek the underlying stochastic feature of interest to yield a robust output and ensure qualitative knowledge.

II. Methods and Materials

A. Credit-Card Fraud Detection: Review of Literature

The cost of financial crimes (globally) was estimated to be about $ 42 billion in 2018. With this, is constantly on the rise – the financial services industry must employ systems that implement innovative fraud mitigation and prevention modes. Many methods for detecting abuse of a technical system [24], [25], are required. Fraud detection seeks to detect cases of fraud from logged data and user behavior [26]. Fraud *management thus* advances a step further to set up preventive measures. Oracle offers real-time detection and correlation capabilities of complex user behavior with use-case management – to result in its early detection and prevention via complex, multi-channel with reduced risk [27]. Fraudsters continue to seek effective means with improved complexity and circumvent border systems, which profile behavior at the point of access [28], and internal hacks that seek to steal client data and defraud valuable clients. Fraud monitoring should offer combined risk monitoring and detection analytics [29]. The system must intelligently correlate event alerts from various channels to offer optimal solutions via early fraud detection of multi-channel, and complex fraud, enhance client protection, and minimize risks [30]–[32].

Fraud is an unlawful act of possessing a valuable asset via intended misrepresentation. It is also associated with criminal cases such as embezzlement, theft, and larceny. It posits that an unknowing victim depends largely on a criminal's bogus claims for gains. It is committed by either an internal/external user. Today, credit cards have not only enhanced their usefulness in financial inclusion, but they have also attracted malicious attacks for gains [33], [34]. With credit cards easily targeted – crimes perpetrated with them are only discovered days afterward. Successful credit-card fraud techniques include (not limited to): (a) card-cloning and acquiring user’s data, and (b) vendors' over-charge without cardholder's awareness [35], [36]. When banks lose money to card fraud, a cardholder is partly or wholly made to pay for such loss via many means that include higher interest rates, and reduced benefits. Thus, it is in both cardholders' and banks' interest to reduce fraudulent acts on a card [17], [37], [38].

In [39], the RBF model used 7 features and the trained RBF recognizes a packet as an attack, it is sent to a filter alarm. Else, it is classified as a normal packet. Profiles were constructed via stream sampling. Results showed that we can: (a) accurately profile packets, and (b) identify anomalies in low false-positive and false-negative. As routers exchange data, they capture key-feats in each packet – allowing them to profile the packets, and increase their rate and confidence in detection. Also [40] posited a distributed change aggregation trees (CATs) detection scheme. It lets the router detect minor shocks in data – which is then investigated and events correlated at the different sessions. The router then proactively terminates the session (if it detects an attack is imminent). In [41], the supervised memetic rule-based model used 7-feats to monitor, inspect and detect packet rates. However, [1] sought to extend the work [41] via deep learning, an unsupervised modular network that captures a packet's key feats used as a profile to help analyze and classify packet patterns in a traffic session as either the normal or a DDoS attack.

B. Data Gathering / Sample Population

Datasets are transactions generated through the Central Bank of Nigeria e-channel having 41,667 records with 15 feats as in Table I, which shows a description of the collected dataset including cardholder and transaction data. We split the dataset into training (70%) and Testing (30%) as in [18], [42], [43].

TABLE I. Dataset Description, Data Types, and Format

Features	Description of Features	DataType	Format
User Name	Account Holder’s Name	Object	abcd
Bank Name	Bank of Account Holder	Object	abcd
NUBAN Account	Nigerian Universal Bank Number e-channel Trans.	Int	1234
Billing Address	Account holder's local bank address of withdrawal, hotel	Object	abcd
Transaction Amount	Amount of transactions adjusted in the bank’s currency	Float	12.34
Transaction Type	Local, International, and/or e-Commerce as type	Object	Abcd
Date/Time	Transaction Date and Time	Float	M:D:Y
Transaction Channel	Channel (payment terminal and/or merchant application)	Object	Abcd
Merchant	Hotels, Restaurants, etc	Object	Abcd
Transaction Gap Time	Duration from last transaction to the current transaction	Float	M:D:Y
Daily Transaction	Daily average transactions performed by a cardholder	Int	1234
Daily Tran. Limit	The daily limit of the amount that cardholders can do daily	Float	12.34
Weekly Transaction	Weekly average transactions performed by the cardholder	Int	1234
Monthly Transaction	Monthly average transactions by the cardholder	Int	1234
Freq. Trans. Types	Average frequency of transactions by cardholder	Int.	1234

C. Parameter / Features Tuning

A critical issue in machine learning is the formatting of data, the selections of feats/parameters of interest to understudy, properly encode the chosen dataset, and tuning the parameters to avoid model overfitting and overtraining to mention a few. Datasets are often rippled with inconsistencies, ambiguities, partial truths, and noise – such that selecting optimal parameters for a model, and encoding it by mapping to the required form a model understands – is a herculean feat and task. To transform our parameters and map them to the dataset, we use the Pandas data type Library as in the listing 1 algorithm [44]–[46].

```
Algorithm 1: Data Description for DeLClustE Algorithm
Input: Features are Selected
Output: Features are Converted to Appropriate Data Type
Initialize DeLClustE with Select Parameters
For Each Selected Parameter do
    if Feature Selected is Non-Numerical then
        Category for the Data Type is Generated
    End if
End For
```

D. Experimental DeLClustE Ensemble

Fig. 1 details our hybrid ensemble leveraging the work of [47]. The ensemble leans on two-components namely: the profile Hidden Markov model and deep-learning neural network. The selected training data forms a cluster of parameters that are passed via a PHMM represented as thus: (a) circles are delete-states for unclassified rules, (b) rectangles as accurately matched states that classify rules into class types, and (c) diamonds are insert-states to update the rules knowledgebase. As PHMM moves between the states, its insert and match states observe the emission state with probabilities corresponding to B. Thus, computes the probabilities via a forward algorithm, and computes the frequency of the number of rules each state emits [48]. Lastly, the delete state lets a PHMM pass through the gaps in the model to reach other emission states. The gaps in the model prevent it from over-fitting and over-parameterization [47], [49], [50].

DNN in its bid to learn, and adapt useful selected parameters via a carefully constructed deep, multilayer net that aims to improve forecast precision. Its hidden layer often transforms [51] data non-linearly and passes it on from a previous layer to its next [52]. With learning handed over to the DNN, [53] stressed that the DNN trains itself using 2-stages namely pre-training, and fine-tuning. It learns and resolves each task posed to it thus: (a) first, it groups all training data into cliques, to find the center point of each clique, (b) it then trains each clique of the DNN [54], learning the various features of each data subset, (c) third, it then applies a test data to previous clique centers to detect the outlier(s) in the pre-trained DNNs, and (d) lastly, it aggregates the output of each DNN as the final result of outliers [40]. A detailed description of the benchmark DNN is described in [55]. Also, the experimental ensemble yields a 3-phase model as in Fig. 1 [56]–[59].

The stages are as thus [14], [52], [53], [60]–[62]:

1) **Step 1**: Separates the data into clusters (train and test). DNN computes cluster centers and uses them as initialization centers to yield test datasets. The data attributes are structured as data points and aligned to meet the classes [63]. The model revises the cluster counts and sigma to improve its performance. The shortest distance between a data point and each cluster center is measured, and a data point's proximity to a cluster classifies it. DNNs use the training sets created by clusters as input. The number of DNNs in training equals the number of cliques. Each DNN consists of 5 layers (input, 2-hidden, softmax, and output). Each training subset is used to train the hidden layer, and the top layer is a 5D output vector. Each training created by the *k*th-clique center is sent back to the *k*th-DNN. And each sub-DNN is trained, and labelled from 1 to k [64], [65].

2) **Step 2**: Generates a k-dataset of data via previous clique center obtained from clusters in Step 1. Test sub-datasets are represented by the letters Test-1 via Test-k [56].

3) **Step 3**: The test dataset is then, fed to the k-sub-DNNs that are trained as in step 1. Each DNN output is combined as the final result to analyze the positive detection [62].

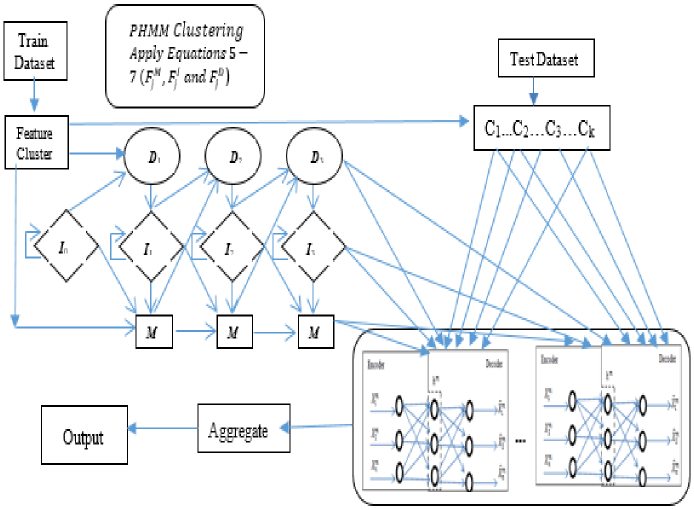

Fig. 1. DeLClustE: deep learning cluster hidden markov model trained deep learning neural network) ensemble.

```
Algorithm 2: The DeLClustE Algorithm
Input: Selected Features, Output: Converted Feature Data Type
Initialize DeLClustE with PHMM; states;
Discrete HMM with Random multinomial draw for each step
K-means cluster fits on k-sub DNNs
For Each Selected Parameter do
    Sample states using Forward Filtering
        Compute Backward Sampling algorithm on states
        Sample each transition parameters
        $T_i = Mult(n_{ij} \setminus T_i)\ Dir\ (T_i \setminus c_k)$
End For Each
```

E. Hyper-Parameter and Ensemble Optimization

A critical issue is that of tuning the parameters with values beyond the ensemble. They ripple across the model as hidden elements (hyper-parameters) [66]–[68] to impact its behavior – via targeted learning to optimize. Thus, we adopt the modular neural network [69]. As the model learns these feats directly via training data, it resolves the issues of over-parameterization, poor generalization, and model over-fitting [1], [70]–[72].

Handling these hyper-parameters is detailed in [57] thus:

1) Learning Rate hyper-feat regulates what weight and how much of it on the network must be modified for gradient loss. A smaller value yields a slower slope. This feat defines how easily a network abandons learn beliefs, in favor of new ones. A small learning rate value implies that a network can quickly distinguish

between important feats and unimportant ones. A faster learning rate allows the net to adapt to change, more easily. To minimize over-fit and overtraining, the learning rate is suitably adjusted.

2) Batch Size defines the size of training used in iteration. There are three-modes namely: (a) batch is when its iteration and epoch values are equal, (b) Mini-batch denotes when the iteration size is greater than its epoch size, and (c) stochastic is when the gradient and network feats, which are updated after each iteration.

An epoch is the number of times when all training values were used to update a weight. A network can be trained in a single step. Training a network in a single pass, on a training dataset – implies that an epoch has been reached or exhausted. Training can span multiple iterations and/or eras. Thus, in batch training – a learning model process all samples simultaneously in one epoch, and update all the weights; While sequential training – adjusts all the weights after a training session.

III. Results and Findings Discussion

A. *Result Findings*

We modeled the network's input layer with one neuron for each parameter to yield a total of 8 neurons; And used two neurons (to represent each possible outcome) for our output layer. The Deep learning parameters include the learning rate, our activation function, the hidden layer structure, and the number of epochs. We used the Rectified Linear Unit Activation Function with 500 epochs (optimal values reached 100, 300, and 500 epochs) – accounting for train convergence time and accuracy). Also, we note that there are no best practices for determining the number of neurons cum hidden layers – and additional hidden layers will give the ensemble capability to undertake more sophisticated functions on the data.

TABLE II. First Hidden Layer Configuration Analysis

Hidden Layers	Precision	Recall	F1	Iteration	Train Loss	Epoch
1	0.94	0.94	0.92	18	1.400	500
2	0.86	0.53	0.63	4	2.230	500
3	0.90	0.84	0.86	16	2.071	500
4	0.92	0.93	0.92	18	1.140	500
5	0.92	0.92	0.90	16	1.779	500
6	0.88	0.91	0.89	7	2.134	500
7	0.91	0.92	0.89	8	2.320	500
8	0.87	0.87	0.87	13	2.006	500
9	0.92	0.92	0.90	8	1.970	500
10	0.92	0.92	0.90	5	1.730	500
11	0.85	0.85	0.85	10	1.540	500
12	0.90	0.84	0.86	15	2.320	500
13	0.91	0.92	0.90	8	1.440	500
14	0.92	0.93	0.90	14	2.160	500
15	0.91	0.91	0.91	5	1.772	500

We choose the number of neurons vis-à-vis the hidden layers via a trial-and-error mode that analyzes the results to achieve its best fit with the least amount of training error. The best number of layers to be used was discovered by first conducting experiments on a single layer with 1-to-15 neurons to determine which produces the highest f-score with the least (constant) amount of training loss time (see Table II)

As in Table III, the addition of a second hidden layer with the greatest number of neurons to generate the highest f-score resulted in the overall best feasible hidden layer arrangement.

TABLE III. Second Hidden Layer Configuration Analysis

Hidden Layers	Precision	Recall	F1	Iteration	Train Loss	Epoch
9, 1	0.91	0.92	0.89	10	1.996	500
9, 2	0.84	0.92	0.88	24	0.281	500
9, 3	0.93	0.93	0.92	11	1.884	500
9, 4	0.92	0.92	0.89	12	1.590	500
9, 5	0.90	0.92	0.90	12	1.731	500
9, 6	0.95	0.94	0.93	14	0.390	500
9, 7	0.93	0.93	0.91	12	1.130	500
9, 8	0.91	0.92	0.91	20	1.929	500
9, 9	0.92	0.93	0.90	13	2.237	500
9, 10	0.94	0.94	0.92	7	1.765	500
9, 11	0.85	0.52	0.62	7	2.010	500
9, 12	0.94	0.94	0.94	6	1.620	500
9, 13	0.93	0.94	0.92	7	1.760	500
9, 14	0.86	0..74	0.79	13	2.059	500
9, 15	0.92	0.92	0.89	8	2.421	500

B. *Discussion of Findings*

To evaluate how well the ensemble performed against known benchmarks, a comparative result(s) is seen in Table IV – with detection accuracies of 0.89 for PHMM, 0.78 for GANN, 0.91 for MNN, 0.96 for DNN, and 0.92 PHMM-DNN respectively. We also have that Fig. 2 shows the mean-time convergence for the various ensembles. We created a total of 22 rules. Table IV shows that rules can effectively identify/detect more than 60-to-82 percent of the cases in the dataset.

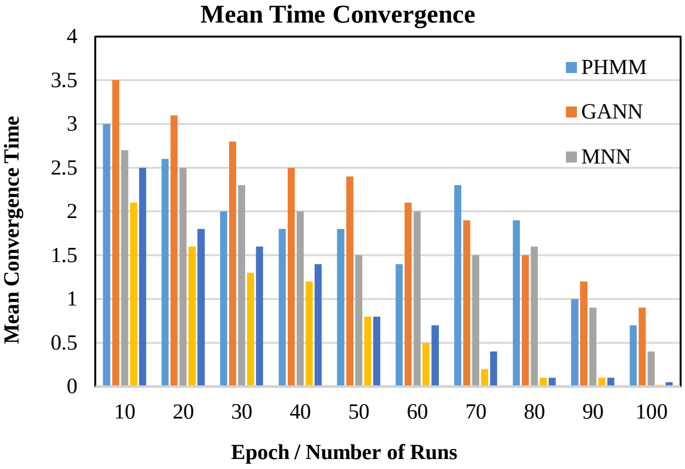

Fig. 2. Mean convergence time for experimental ensemble.

TABLE IV. CONFIGURATION ANALYSIS

Ensemble	Precision	Redundancy	Recall	F1	Average Support	Cost Estimate
DNN	0.92	0.52	0.94	0.92	12.449	140
PHMM	0.89	0.71	0.83	0.83	11.410	130
MNN	0.91	0.56	0.92	0.89	11.411	140
GANN	0.78	0.79	0.74	0.92	11.408	120
DeLClustE	0.96	0.52	1.00	0.97	12.500	140

That is, the test phase of the model with 12,500 records reveals that we accurately identified the majority of the models, 11,411 benign cases as agreed by [73], [74]. The result showed 11,410 benign threats from the test dataset are correctly grouped (i.e. true-positives). The result showed that 31-detected cases were erroneously labeled and agreed with [75]–[77] as false-positive; Also, 776 wrongly detected threats (i.e. false-negative) and 283-correctly recognized malicious instances labeled as true-negative. Thus, for true-positive cases, the model predicted positive (correctly) and also predicted negative (correctly) for true-negative cases. Conversely, sensitivity and specificity rates were computed with standardized tests for our test data [78], [79]. These proved, to be more efficient.

IV. CONCLUSION

We created a total of 22 rules, with classification accuracy range and fitness [0.6, 0.82] for the top rules (i.e. 60% of generated rules can sufficiently categorize the dataset). Thus, the ensemble effectively/correctly identifies fraudulent transactions and simultaneously improves the generality of rules to allow new datasets and their associated produced rules, to be added to the knowledgebase. Detection often filters all requests on a network, analyses all to separate compromised clients from those that are uncompromised, and also, provides security measures as appropriate actions. The performance of these ensembles may be hampered by error rates for erroneously classified and misidentified data points generated by the scheme and/or model.

Through trade-offs between the frequency of false positives and false negatives, an ideal approach correctly classifies all requests with nearly zero error rates of false positives or false negatives. With the increasing trend of intrusion threats and activities, it is critical to develop new methods and updated security monitoring systems that provide a high chance of detection and timely warning of intruder attacks. The goal of this research is to adapt a hybrid ensemble to monitor card-holder transaction flow patterns on a network, predict possible fraudster and adversary behaviors, and boost the effectiveness of banking platform (e-channel) network security when and if the level of threat changes.

REFERENCES

[1] A. A. Ojugo and R. E. Yoro, “Forging a deep learning neural network intrusion detection framework to curb the distributed denial of service attack,” Int. J. Electr. Comput. Eng., vol. 11, no. 2, pp. 1498–1509, 2021, doi: 10.11591/ijece.v11i2.pp1498-1509.

[2] C. L. Udeze, I. E. Eteng, and A. E. Ibor, “Application of Machine Learning and Resampling Techniques to Credit Card Fraud Detection,” J. Niger. Soc. Phys. Sci., p. 769, Aug. 2022, doi: 10.46481/jnsps.2022.769.

[3] M. Dadkhah, T. Sutikno, J. M. Davarpanah, and D. Stiawan, “An Introduction to Journal Phishings and Their Detection Approach,” TELKOMNIKA, vol. 13, no. 2, p. 373, Jun. 2015, doi: 10.12928/telkomnika.v13i2.1436.

[4] Y. Abakarim, M. Lahby, and A. Attioui, “An Efficient Real Time Model For Credit Card Fraud Detection Based On Deep Learning,” in Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, Oct. 2018, pp. 1–7. doi: 10.1145/3289402.3289530.

[5] S. M. Albladi and G. R. S. Weir, “User characteristics that influence judgment of social engineering attacks in social networks,” Human-centric Comput. Inf. Sci., vol. 8, no. 1, p. 5, Dec. 2018, doi: 10.1186/s13673-018-0128-7.

[6] V. Filippov, L. Mukhanov, and B. Shchukin, “Credit card fraud detection system,” in 2008 7th IEEE International Conference on Cybernetic Intelligent Systems, Sep. 2008, pp. 1–6. doi: 10.1109/UKRICIS.2008.4798919.

[7] A. Algarni, Y. Xu, and T. Chan, “An empirical study on the susceptibility to social engineering in social networking sites: the case of Facebook,” Eur. J. Inf. Syst., vol. 26, no. 6, pp. 661–687, Nov. 2017, doi: 10.1057/s41303-017-0057-y.

[8] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, “Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection,” IEEE Access, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[9] I. A. Anderson and W. Wood, “Habits and the electronic herd: The psychology behind social media’s successes and failures,” Consum. Psychol. Rev., vol. 4, no. 1, pp. 83–99, Jan. 2021, doi: 10.1002/arcp.1063.

[10] F. O. Aghware, R. E. Yoro, P. O. Ejeh, C. Odiakaose, F. U. Emordi, and A. A. Ojugo, “Sentiment Analysis in Detecting Sophistication and Degradation Cues in Malicious Web Contents,” Kongzhi yu Juece/Control Decis., vol. 38, no. 01, pp. 653–665, 2023.

[11] A. E. Ibor, E. B. Edim, and A. A. Ojugo, “Secure Health Information System with Blockchain Technology,” J. Niger. Soc. Phys. Sci., vol. 5, no. 992, pp. 1–8, 2023, doi: 10.46481/jnsps.2022.992.

[12] I. Correia, F. Fournier, and I. Skarbovsky, “The uncertain case of credit card fraud detection,” in Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, Jun. 2015, pp. 181–192. doi: 10.1145/2675743.2771877.

[13] Y. Lucas et al., “Multiple perspectives HMM-based feature engineering for credit card fraud detection,” in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Apr. 2019, pp. 1359–1361. doi: 10.1145/3297280.3297586.

[14] S. S. Verma et al., “Collective feature selection to identify crucial epistatic variants,” BioData Min., vol. 11, no. 1, p. 5, Dec. 2018, doi: 10.1186/s13040-018-0168-6.

[15] L. De Kimpe, M. Walrave, W. Hardyns, L. Pauwels, and K. Ponnet, “You’ve got mail! Explaining individual differences in becoming a phishing target,” Telemat. Informatics, vol. 35, no. 5, pp. 1277–1287, Aug. 2018, doi: 10.1016/j.tele.2018.02.009.

[16] G. M. Friesen, T. C. Jannett, M. A. Jadallah, S. L. Yates, S. R. Quint, and H. T. Nagle, “A comparison of the noise sensitivity of nine QRS detection algorithms,” IEEE Trans. Biomed. Eng., vol. 37, no. 1, pp. 85–98, 1990, doi: 10.1109/10.43620.

[17] A. Artikis et al., “A Prototype for Credit Card Fraud Management,” in Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, Jun. 2017, pp. 249–260. doi: 10.1145/3093742.3093912.

[18] C. Li, N. Ding, H. Dong, and Y. Zhai, “Application of Credit Card Fraud Detection Based on CS-SVM,” Int. J. Mach. Learn. Comput., vol. 11, no. 1, pp. 34–39, 2021, doi: 10.18178/ijmlc.2021.11.1.1011.

[19] S. Goel, K. Williams, and E. Dincelli, “Got Phished? Internet Security and Human Vulnerability,” J. Assoc. Inf. Syst., vol. 18, no. 1, pp. 22–44, Jan. 2017, doi: 10.17705/1jais.00447.

[20] T. Halevi, J. Lewis, and N. Memon, “A pilot study of cyber security and privacy related behavior and personality traits,” in Proceedings of the 22nd International Conference on World Wide Web, May 2013, pp. 737–744. doi: 10.1145/2487788.2488034.

[21] A. A. Ojugo and O. D. Otakore, “Intelligent cluster connectionist recommender system using implicit graph friendship algorithm for social networks,” IAES Int. J. Artif. Intell., vol. 9, no. 3, p. 497~506, 2020, doi: 10.11591/ijai.v9.i3.pp497-506.

[22] Y. Gao, S. Zhang, J. Lu, Y. Gao, S. Zhang, and J. Lu, “Machine Learning for Credit Card Fraud Detection,” in Proceedings of the 2021 International Conference on Control and Intelligent Robotics, Jun. 2021, pp. 213–219. doi: 10.1145/3473714.3473749.

[23] M. Gratian, S. Bandi, M. Cukier, J. Dykstra, and A. Ginther, “Correlating human traits and cyber security behavior intentions,” Comput. Secur., vol. 73, pp. 345–358, Mar. 2018, doi: 10.1016/j.cose.2017.11.015.

[24] M. I. Akazue, A. A. Ojugo, R. E. Yoro, B. O. Malasowe, and O. Nwankwo, “Empirical evidence of phishing menace among undergraduate smartphone users in selected universities in Nigeria,” Indones. J. Electr. Eng. Comput. Sci., vol. 28, no. 3, pp. 1756–1765, Dec. 2022, doi: 10.11591/ijeecs.v28.i3.pp1756-1765.

[25] R. E. Yoro, F. O. Aghware, M. I. Akazue, A. E. Ibor, and A. A. Ojugo, “Evidence of personality traits on phishing attack menace among selected university undergraduates in Nigerian,” Int. J. Electr. Comput. Eng., vol. 13, no. 2, p. 1943, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1943-1953.

[26] A. Abbasi, F. M. Zahedi, and Y. Chen, “Phishing susceptibility: The good, the bad, and the ugly,” in 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Sep. 2016, pp. 169–174. doi: 10.1109/ISI.2016.7745462.

[27] J. R. Amalraj and R. Lourdusamy, “A Novel Distributed Token-Based Access Control Algorithm Using A Secret Sharing Scheme for Secure Data Access Control,” Int. J. Comput. Networks Appl., vol. 9, no. 4, p. 374, Aug. 2022, doi: 10.22247/ijcna/2022/214501.

[28] A. A. Ojugo, M. I. Akazue, P. O. Ejeh, C. Odiakaose, and F. U. Emordi, “DeGATraMoNN : Deep Learning Memetic Ensemble to Detect Spam Threats via a Content-Based Processing,” Kongzhi yu Juece/Control Decis., vol. 38, no. 01, pp. 667–678, 2023.

[29] A. A. Ojugo, C. O. Obruche, and A. O. Eboka, “Quest For Convergence Solution Using Hybrid Genetic Algorithm Trained Neural Network Model For Metamorphic Malware Detection,” ARRUS J. Eng. Technol., vol. 2, no. 1, pp. 12–23, Nov. 2021, doi: 10.35877/jetech613.

[30] M. Barlaud, A. Chambolle, and J.-B. Caillau, “Robust supervised classification and feature selection using a primal-dual method,” Feb. 2019.

[31] E. R. Altman, “Synthesizing Credit Card Transactions,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.03033

[32] A. A. Ojugo and A. O. Eboka, “Empirical Bayesian network to improve service delivery and performance dependability on a campus network,” IAES Int. J. Artif. Intell., vol. 10, no. 3, p. 623, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp623-635.

[33] I. Benchaji, S. Douzi, B. El Ouahidi, and J. Jaafari, “Enhanced credit card fraud detection based on attention mechanism and LSTM deep model,” J. Big Data, vol. 8, no. 1, p. 151, Dec. 2021, doi: 10.1186/s40537-021-00541-8.

[34] M. Fatahi, M. Ahmadi, A. Ahmadi, M. Shahsavari, and P. Devienne, “Towards an spiking deep belief network for face recognition application,” in 2016 6th International Conference on Computer and Knowledge Engineering (ICCKE), Oct. 2016, pp. 153–158. doi: 10.1109/ICCKE.2016.7802132.

[35] E. Ileberi, Y. Sun, and Z. Wang, “A machine learning based credit card fraud detection using the GA algorithm for feature selection,” J. Big Data, vol. 9, no. 1, p. 24, Dec. 2022, doi: 10.1186/s40537-022-00573-8.

[36] M. I. Akazue, R. E. Yoro, B. O. Malasowe, O. Nwankwo, and A. A. Ojugo, “Improved services traceability and management of a food value chain using block-chain network : a case of Nigeria,” Indones. J. Electr. Eng. Comput. Sci., vol. 29, no. 3, pp. 1623–1633, 2023, doi: 10.11591/ijeecs.v29.i3.pp1623-1633.

[37] M. Laavanya and V. Vijayaraghavan, “Real Time Fake Currency Note Detection using Deep Learning,” Int. J. Eng. Adv. Technol., vol. 9, no. 1S5, pp. 95–98, 2019, doi: 10.35940/ijeat.a1007.1291s52019.

[38] R. Broadhurst, K. Skinner, N. Sifniotis, and B. Matamoros-Macias, “Cybercrime Risks in a University Student Community,” SSRN Electron. J., no. May, 2018, doi: 10.2139/ssrn.3176319.

[39] R. Brause, F. Hamker, and J. Paetz, “Septic Shock Diagnosis by Neural Networks and Rule Based Systems,” 2002, pp. 323–356. doi: 10.1007/978-3-7908-1788-1_12.

[40] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, “A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks,” Sensors, vol. 16, no. 10, p. 1701, Oct. 2016, doi: 10.3390/s16101701.

[41] A. A. Ojugo, A. O. Eboka, E. O. Okonta, R. E. Yoro, and F. O. Aghware, “Genetic Algorithm Rule-Based Intrusion Detection System (GAIDS),” J. Emerg. Trends Comput. Inf. Syst., vol. 3, no. 8, pp. 1182–1194, 2012, [Online]. Available: http://www.cisjournal.org

[42] S. V. S. . Lakshimi and S. D. Kavila, “Machine Learning for Credit Card Fraud Detection System,” Int. J. Appl. Eng. Res., vol. 15, no. 24, pp. 16819–16824, 2018, doi: 10.1007/978-981-33-6893-4_20.

[43] A. Jayatilaka, N. A. G. Arachchilage, and M. A. Babar, “Falling for Phishing: An Empirical Investigation into People’s Email Response Behaviors,” arXiv Prepr. arXiv …, no. Fbi 2020, pp. 1–17, 2021.

[44] L. E. Mukhanov, “Using bayesian belief networks for credit card fraud detection,” Proc. IASTED Int. Conf. Artif. Intell. Appl. AIA 2008, no. February 2008, pp. 221–225, 2008.

[45] N. Rtayli and N. Enneya, “Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization,” J. Inf. Secur. Appl., vol. 55, p. 102596, Dec. 2020, doi: 10.1016/j.jisa.2020.102596.

[46] D. Huang, Y. Lin, Z. Weng, and J. Xiong, “Decision Analysis and Prediction Based on Credit Card Fraud Data,” in The 2nd European Symposium on Computer and Communications, Apr. 2021, pp. 20–26. doi: 10.1145/3478301.3478305.

[47] A. A. Ojugo and E. O. Ekurume, “Deep Learning Network Anomaly-Based Intrusion Detection Ensemble For Predictive Intelligence To Curb Malicious Connections: An Empirical Evidence,” Int. J. Adv. Trends Comput. Sci. Eng., vol. 10, no. 3, pp. 2090–2102, Jun. 2021, doi: 10.30534/ijatcse/2021/851032021.

[48] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, “Real Time Data-Driven Approaches for Credit Card Fraud Detection,” in Proceedings of the 2018 International Conference on E-Business and Applications - ICEBA 2018, 2018, pp. 6–9. doi: 10.1145/3194188.3194196.

[49] A. A. Ojugo and O. Nwankwo, “Spectral-Cluster Solution For Credit-Card Fraud Detection Using A Genetic Algorithm Trained Modular Deep Learning Neural Network,” JINAV J. Inf. Vis., vol. 2, no. 1, pp. 15–24, Jan. 2021, doi: 10.35877/454RI.jinav274.

[50] X. E. Pantazi, D. Moshou, T. Alexandridis, R. L. Whetton, and A. M. Mouazen, “Wheat yield prediction using machine learning and advanced sensing techniques,” Comput. Electron. Agric., vol. 121, pp. 57–65, Feb. 2016, doi: 10.1016/j.compag.2015.11.018.

[51] A. A. Ojugo and D. A. Oyemade, “Boyer moore string-match framework for a hybrid short message service spam filtering technique,” IAES Int. J. Artif. Intell., vol. 10, no. 3, pp. 519–527, 2021, doi: 10.11591/ijai.v10.i3.pp519-527.

[52] G. Sasikala et al., “An Innovative Sensing Machine Learning Technique to Detect Credit Card Frauds in Wireless Communications,” Wirel. Commun. Mob. Comput., vol. 2022, pp. 1–12, Jun. 2022, doi: 10.1155/2022/2439205.

[53] H. Tingfei, C. Guangquan, and H. Kuihua, “Using Variational Auto Encoding in Credit Card Fraud Detection,” IEEE Access, vol. 8, pp. 149841–149853, 2020, doi: 10.1109/ACCESS.2020.3015600.

[54] A. A. Ojugo and R. E. Yoro, “Extending the three-tier constructivist learning model for alternative delivery: ahead the COVID-19 pandemic in

Nigeria," Indones. J. Electr. Eng. Comput. Sci., vol. 21, no. 3, p. 1673, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1673-1682.

[55] D. Wang, B. Chen, and J. Chen, "Credit card fraud detection strategies with consumer incentives," Omega, vol. 88, pp. 179–195, Oct. 2019, doi: 10.1016/j.omega.2018.07.001.

[56] A. Seleznyov, An Anomaly Intrusion Detection System Based on Intelligent User Recognition An Anomaly Intrusion Detection System Based on Intelligent User Recognition. 2002.

[57] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), Mar. 2018, pp. 1–6. doi: 10.1109/ICNSC.2018.8361343.

[58] Maya Gopal P S and Bhargavi R, "Selection of Important Features for Optimizing Crop Yield Prediction," Int. J. Agric. Environ. Inf. Syst., vol. 10, no. 3, pp. 54–71, Jul. 2019, doi: 10.4018/IJAEIS.2019070104.

[59] P. . Maya Gopal and Bhargavi R, "Feature Selection for Yield Prediction Using BORUTA Algorithm," Int. J. Pure Appl. Math., vol. 118, no. 22, pp. 139–144, 2018.

[60] M. Zareapoor and P. Shamsolmoali, "Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier," Procedia Comput. Sci., vol. 48, pp. 679–685, 2015, doi: 10.1016/j.procs.2015.04.201.

[61] Q. Li et al., "An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis," Comput. Math. Methods Med., vol. 2017, pp. 1–15, 2017, doi: 10.1155/2017/9512741.

[62] P. Moodley, D. C. S. Rorke, and E. B. Gueguim Kana, "Development of artificial neural network tools for predicting sugar yields from inorganic salt-based pretreatment of lignocellulosic biomass," Bioresour. Technol., vol. 273, pp. 682–686, Feb. 2019, doi: 10.1016/j.biortech.2018.11.034.

[63] A. O. Eboka and A. A. Ojugo, "Mitigating technical challenges via redesigning campus network for greater efficiency, scalability and robustness: A logical view," Int. J. Mod. Educ. Comput. Sci., vol. 12, no. 6, pp. 29–45, 2020, doi: 10.5815/ijmecs.2020.06.03.

[64] V. Vijayaraghavan and M. Laavanya, "Vehicle Classification and Detection using Deep Learning," Int. J. Eng. Adv. Technol., vol. 9, no. 1S5, pp. 24–28, 2019, doi: 10.35940/ijeat.a1006.1291s52019.

[65] I. Sohony, R. Pratap, and U. Nambiar, "Ensemble learning for credit card fraud detection," in Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, Jan. 2018, pp. 289–294. doi: 10.1145/3152494.3156815.

[66] M. Zanin, M. Romance, S. Moral, and R. Criado, "Credit Card Fraud Detection through Parenclitic Network Analysis," Complexity, vol. 2018, pp. 1–9, 2018, doi: 10.1155/2018/5764370.

[67] K. Kuwata and R. Shibasaki, "Estimating crop yields with deep learning and remotely sensed data," in 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Jul. 2015, pp. 858–861. doi: 10.1109/IGARSS.2015.7325900.

[68] Z. Karimi, M. Mansour Riahi Kashani, and A. Harounabadi, "Feature Ranking in Intrusion Detection Dataset using Combination of Filtering Methods," Int. J. Comput. Appl., vol. 78, no. 4, pp. 21–27, Sep. 2013, doi: 10.5120/13478-1164.

[69] G. Behboud, "Reasoning using Modular Neural Network," Towar. Data Sci., vol. 34, no. 2, pp. 12–34, 2020.

[70] S. Nosratabadi, F. Imre, K. Szell, S. Ardabili, B. Beszedes, and A. Mosavi, "Hybrid Machine Learning Models for Crop Yield Prediction," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2005.04155

[71] S. Khaki and L. Wang, "Crop Yield Prediction Using Deep Neural Networks," Front. Plant Sci., vol. 10, May 2019, doi: 10.3389/fpls.2019.00621.

[72] S. Khaki, L. Wang, and S. V. Archontoulis, "A CNN-RNN Framework for Crop Yield Prediction," Front. Plant Sci., vol. 10, Jan. 2020, doi: 10.3389/fpls.2019.01750.

[73] D. Nahavandi, R. Alizadehsani, A. Khosravi, and U. R. Acharya, "Application of artificial intelligence in wearable devices: Opportunities and challenges," Comput. Methods Programs Biomed., vol. 213, p. 106541, Jan. 2022, doi: 10.1016/j.cmpb.2021.106541.

[74] H. J. Parker and S. V. Flowerday, "Contributing factors to increased susceptibility to social media phishing attacks," SA J. Inf. Manag., vol. 22, no. 1, Jun. 2020, doi: 10.4102/sajim.v22i1.1176.

[75] Y. Gao, S. Zhang, and J. Lu, "Machine learning for credit card fraud detection," in Proceedings of the 2021 1st International Conference on Control and Intelligent Robotics, 2021, pp. 213–219. doi: 10.1145/3473714.3473749.

[76] D. Zhang, B. Bhandari, and D. Black, "Credit Card Fraud Detection Using Weighted Support Vector Machine," Appl. Math., vol. 11, no. 12, pp. 1275–1291, 2020, doi: 10.4236/am.2020.1112087.

[77] O. V. Lee et al., "A malicious URLs detection system using optimization and machine learning classifiers," Indones. J. Electr. Eng. Comput. Sci., vol. 17, no. 3, p. 1210, Mar. 2020, doi: 10.11591/ijeecs.v17.i3.pp1210-1214.

[78] R. E. Yoro, F. O. Aghware, B. O. Malasowe, O. Nwankwo, and A. A. Ojugo, "Assessing contributor features to phishing susceptibility amongst students of petroleum resources varsity in Nigeria," Int. J. Electr. Comput. Eng., vol. 13, no. 2, p. 1922, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1922-1931.

[79] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification," Int. J. Inf. Manag. Data Insights, vol. 1, no. 2, p. 100048, Nov. 2021, doi: 10.1016/j.jjimei.2021.100048.