Framework Design for Statistical Fraud Detection

A.A. Ojugo., A.O. Eboka., R.E. Yoro., M.O. Yerokun and F.N. Efozia

***Abstract*—**An unstable economy is rife with fraud. Perpetrated on customers, it ranges from employees' internal abuse to large fraud via high-value contracts cum control breaches that impose serious consequences to biz. Loyal employees may not perpetrate fraud if not for societal pressures and economic recession with its rationalization that they have bills to pay and children to feed. Thus, the need for financial institutions to embark on effective measures via schemes that will aids both fraud prevention and detection. Study proposes genetic algorithm trained neural net model to accurately classify credit card transactions. Compared, the model used a rule-based system to provide it with startup solution and it has a fraud catching rate of 91% with a consequent, false alarm rate of 9%. Its convergence time is found to depend on how close the initial solution space is to the fitness function, and for the recombination and mutation rates applied.

***Keywords*—**Classifiers, fitness function, stochastic, mutation, recombination, products, credit-card, transactions

I. Introduction

FRAUD is illegal acquisition of valuable data or resources via willful misrepresentation. Others view it as a criminal act involving larceny, theft and embezzlement. Legally, it is a state where a criminal makes a false, material statement and an unsuspecting victim rely such a statement; while, the criminal benefits from the entire process (Ojugo et al, 2014). It is perpetrated mainly by internal member of an organization and also those external to it. Fraud benefits an organization, its part or internal/external persons to the organization (Marane, 2011) and it has been classified into types as in fig. 1; We note also that fraud involves all products of financial transactions, and grouped as in fig 2 broadly into: (a) ***cheque/check*** fraud involves use of counterfeit or altered checks via forgery of an account holder signature, and (b) C***ybercrimes*** involves use of ***key logging*** (here, keys struck as a user gains access into his system is tracked in a covert manner so that the user is unaware that his action are monitored), and ***hacking*** (intruder gains access of user's PC without their permission). They gain access via a user's USB or the user is a victim of a virus attack via web-surfing, email attachments and/or online purchase. Example of malware techniques includes (Ojugo et al, 2014):

a. **Trojan-Horse** is a form of malware (virus), hidden in a file, program, free online games downloaded also called shareware, hyper- or related links, emails attachments and screensavers.
b. **Spyware** collects bits of user data at a time without the knowledge of an unsuspecting user. Its presence is hidden and difficult to detect. It is prevented by noting the sites visited, and reading emails sent before following any link.
c. **Phishing** attempts to acquire sensitive/confidential details from account holders, masquerading as trustworthy entity in an electronic communication. Perpetrated via online-payment, social and auction sites, phishing lures users via emails or instant messaging into fake site with deceptive login and hyperlink to real sites. Similarly, voice phishing (***vishing***) are attacks in which account holder is contacted mainly by phone, to check if their account status is been compromised. Rather than refer to a site, the unsuspecting customer is redirected to call a toll-free number.

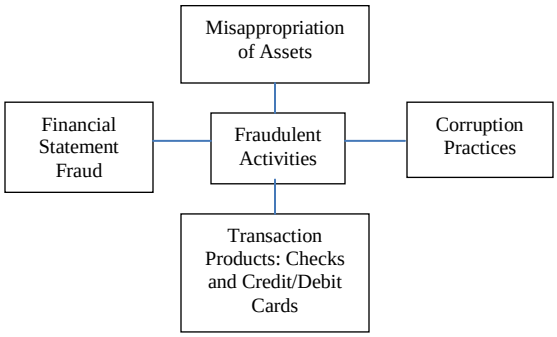

Fig. 1: Types of Fraud and Fraudulent Activities

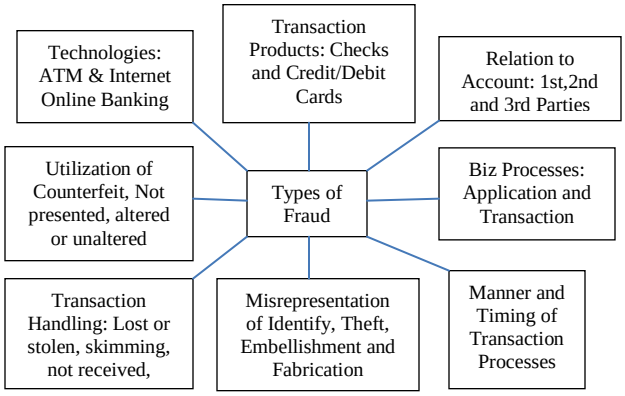

Fig. 2: Forms of Transaction Product Fraud (Delamaire et al, 2009)

A. Fraud Prevention and Detection

When financial institutions lose money to fraud, credit card (account) holders (partially, if not wholly) pay for such lose via reduced benefits, higher interest rate and membership fees. It is to the interest of both the financial institution and her customers to reduce fraud on transaction products. This makes imperative and imminent that financial institutions embark on early detection methods, to savor the situation.

Fraud prevention describes measures to stop the occurrence of fraud in the first place. Measures include elaborate designs, fluorescent fibers, multi-tone drawings, watermarks, laminated metal strips, holographs on banknotes, personal identification numbers for bankcards, security for credit card transactions, password on PCs, Subscriber Identity Module card for mobile phones and telephone bank accounts. With no perfect system, a tradeoff is struck between its inconvenience and expense to a customer and effectiveness of the system to the entire financial process. Thus, fraud detection emerges if its prevention fails, and it involves ***techniques*** for identifying fraud as quick as possible, once it has been perpetrated. In practice, detection is

A.A. Ojugo: Dept. of Math/Computer Sci., Federal University of Petroleum Resources, Effurun, (ojugo.arnold@fupre.edu.ng)

used continuous, as we may be unaware when our prevention technique in place, has failed (Bolton and Hand, 2002).

No matter how carefully we guard/prevent credit card fraud and not beyond mistakes, our confidential details may again be stolen or lost, to end in the wrong hands. Then, we must be able to detect, as soon as possible, that and when fraud is being perpetrated. Fraud detection is a continuously evolving discipline as intruders evolve their strategies to hack into a known detection system. With new criminal minded intruders constantly entering the field, many will however be unaware of previous fraud detection methods that have been successful as they adopt varying strategies. This will however, lead to identifiable frauds and implies that both the earlier and latest detection tools need to be applied always. This also implies that an up-to-date is required to monitor and detect intrusion (Delamaire et al, 2009; Bolton and Hand, 2002).

B. Statistical Fraud Detection

Various statistical methods have been successfully employed in fraud detection as tools. Though diverse in their different applications in both size and type, they have common themes. These tools aim to compare the observed (***historic***) data with its expected (***computed***) data-values. The expected value can be derived cum interpreted in varied ways, depending on the context as they may be single numerical summaries of some aspect of a dataset (feats) behavior with often simple graphical summaries in which an anomaly is readily apparent. They can also be shown as a more complex (multivariate) behavior of data profiles. Such profiles are based on past behavior of the system understudied (e.g. how an account has been previously used), or be extrapolated from other similar systems (Bolton and Hand, 2002). Events are further complicated in that, a given user may behave in a fraudulent manner at some point and not at other times (Delamaire et al, 2009) resulting to true-negatives and false-positive results in fraud detection.

Statistical fraud detection methods may be ***supervised*** or ***unsupervised***. In supervised, samples of both fraudulent and non-fraudulent dataset records are used to construct the model to allow it classify new observations into one of two classes. It requires precise and distinct feats of the original dataset used to build the model, for its classification of the observations into both (***true***) classes as well as examples of both classes. Thus, it can only be used to detect frauds of a type which have previously occurred. Conversely, ***unsupervised*** mode simply seeks those accounts, customers and so forth which are most dissimilar from the norm. These are further examined to detect outliers. Outliers are a basic form of non-standard observation. Such tools will aim at checking data quality as well as separate between false-positives (detection of accidental errors) from true-negatives (detection of deliberately falsified data or data that accurately describe fraudulent pattern). Statistical method alone cannot ascertain if fraud is committed. Such analysis should be viewed as alerting us to anomalous observations that are more likely to be fraudulent than others, so that it can then be investigated in more detail. Thus, the objective of the statistical analysis can be seen as to return a *suspicion score* (higher score is more suspicious than a lower score). Since there are varied ways in which fraud is perpetrated, there also exists many different ways to compute suspicion scores (Bolton and Hands, 2002).

Suspicion scores is computed for each record in database (for each customer with a bank account or credit card, for each owner of a mobile phone, for each desktop computer and so on), and these can be updated as time progresses. These scores are then ranked and investigative attention focused on those with highest scores or on those that exhibit a sudden increase. With cost as an issue here, given that it is too expensive to undertake a detailed investigation of all records, the algorithm investigates only those thought most likely to be fraudulent.

The main ***idea*** here is to employ hybrid genetic algorithm trained neural network model with rule-based preprocessor in computing the suspicion score, as means for fraud detection. This method will seek a solution to this complex and dynamic task, for which conventional method do not yield complete, cost-effective solutions. It analyzes historic data, investigating data features of interest by computing statistically-dependent underlying probabilities between observed and predicted data, to simulate tractability, low-cost, robustness and effective results, with high tolerance to uncertainties, ambiguity, partial truth, imprecision and noise that may have been applied along with its input data.

II. Statement of Problem

The problem statements are as follows:

1. Exchange of idea in fraud detection is often limited since it is unwise to describe in public domain (fraud detection techniques in great detail). It will further equip intruders with adequate information required to evade detection. Thus, we employ statistical fraud detection method and heuristics as in Section III.
2. The complex, dynamic and chaotic nature of fraud, and its range of complications as providing a backdoor to allow for other crime makes imperative and critical, early and accurate detection**.** Supervised detection alone via careful monitoring and management of network is insufficient as intruders often evade such as it often yields inconclusive results for ***unknown*** inputs. This leads to increased rate of false-positives and true-negatives. Our proposed model will effectively classify fraudulent from non-fraudulent activities using soft-computing heuristics as in Section IV.
3. Unavailability of fraud datasets and its censored results – makes fraud detection techniques and studies difficult to assess. Dataset also consist of ambiguities, imprecision, noise and impartial truth that must be resolved via robust search in the bid to classify observations and expected values effectively as in Section III and IV.
4. Classification (as resolved in Section IV) using predictive models is a complex and difficult task due to the chaotic and dynamic nature of ***unsupervised*** evolutionary models. The model also need to resolve effectively and efficiently, statistical dependences and conflict imposed on the model by dataset used in approximating the data feats of interest.
5. Use of hill-climbing methods often has speed constraint imposed on it as the solutions are often trapped at local maxima. This is resolved with hybridization of statistical methods as in Section III/IV. Also, search for optimal via evolutionary heuristics can be quite cumbersome (though no one method yields better optimal than hybrids). Model must also resolve the statistical dependencies imposed on

it by hybridization. The proposed model resolves this in Section III and IV.

6. Search for optimal solution, may also yield overtraining and over-fitting of the model (resolved in Section IV) as it aims to find underlying probability of the data feat(s) of interest. Also, improper selection of parameters may also lead to over-parameterization (resolved in Section V).
7. Some model aim at a single suspicion score to globally classify statistical fraud. Studies show however, that some cases may be a result of true-negatives and false-positives scores as resolved in Section IV.

Proposed genetic algorithm trained neural network will seek to use unsupervised (improved) classification method that will help propagate observed data in model as it seeks data feats of interest to yield an output (Ojugo et al, 2013). Evolutionary models have been successfully applied to enhance accurate prediction in its search for optimal solution, chosen from a set of possible solution space, to yield an output that is guaranteed of high quality and void of ambiguities. These models, further tuned can become robust and perform quantitative processing to ensure qualitative knowledge and experience, as its new language (Heppner and Grenander, 1990).

III. The Proposed Hybrid Model

Proposed model hinges on 3-basic frameworks as in fig 1 – further explained as:

A. Rule-Based Preprocessor

We employ the rule-based system for 3 reasons: (a) it serves as benchmark to measure how well other heuristics perform on comparison, (b) a simplified version yields a sensible solution and generation of rules in the model; Rather, than choosing completely random points swaps and mutations. Thus, greatly improves the proportion of moves accepted, and (c) used as preprocessor to other models, it yield a good starting solution with minimized false-positive cum true-negatives. The rule-based model consists of heuristics and conventional recursion routines to assist in carrying out fraud detection classification, as suited for the problem domain at hand where the following holds (Saleh Elmohamed et al, 1998; Michalewicz, 1998):

a. The Account Data Structure of each account holder in an institution to hold values of each account holder.
b. The Card data structure of cards issued and their types, capable of being used across E-Tranzact and InterSwitch platforms for online banking.
c. Data structure for time periods to keep track of which, when and what transaction took place.
d. Fraud data structure of each time-stamped data containing a knowledgebase of historic data classified into fraudulent and non-fraudulent transactions so as to minimize true-negatives and false positives.

The model's basics function is: given data files of fraudulent and non-fraudulent classes, user account details, time-stamped transactions for account holders, account holders transaction-to-credit card details matrix, the inclusion data that allows for further classification of fraudulent and non-fraudulent dataset. Even when administrator maintains transactions in time, it should reflect classified as it has been noted that fraud results mainly from internal breaches, employees and tips. Using these structures above, the system builds an internal database (knowledgebase) to perform the computation of the suspicion score as well as classification of transaction into classes. It involves a number of essential sub-processes such as checking all time-stamped transaction, entry access point by the account holder as well as conflict resolution that arise in transactions (such as rollback to debit an account holder via online access from credit card) amongst other transaction processes (Ojugo et al, 2014).

The rule-based model is an iterative approach that allows rule generation as its basic procedure for each move as thus. Model first scans through all currently unassigned transactions by choosing the transactions in order of account holder's details, the transaction type and then computes suspicion score for each transaction sampled. It then attempts to assign these transactions between fraudulent and non-fraudulent classes by satisfying the rules governing their consequent classification. Only transactions that satisfy all the rule constraints can be classified. Model then searches through classified (fraudulent and non-fraudulent) knowledgebase of transaction and selects those with higher suspicion score. Selecting threshold values for defining what is considered a high score for each case is a subjective procedure (though, it is also a more straightforward approach to choose a reasonable value). Thus, the rule-based model yields a partial output, as it is unable to assign the given transactions into their varying classes (Ojugo et al, 2014).

B. Genetic Algorithm (GA)

GA is inspired by Darwinian genetic evolution (survival of fittest) consists of population (data) chosen for selection with potential solutions to a specific task. Each potential solution is an individual for which optimal is found using four operators: initialize, select, crossover and mutation (Coello et al, 2004 and Reynolds, 1994). Individuals with genes close to optimal are fit. Its fitness function determines how close an individual is to the optimal solution. Its operators are (Ojugo et al, 2012):

a. Initialize encodes data into forms suitable for selection. Each encodings type used has its merit. Binary encoding is computationally more expensive. Decimal encoding has greater diversity in chromosome and greater variance of pools generated. Float-point encoding or its combination is more efficient. The *fitness* function evaluates how close a solution is to its optimal – after which they are chosen for reproduction. If solution is found, function is *good*; else, *bad* and deselected for crossover. Fitness function is the only part with knowledge of the task, and if more solutions are found, the higher its fitness value.
b. Selection: With best fit data chosen to mate, the larger the number selected, the better the chances of yielding fitter data. This continues until one is chosen, from the last 2 or 3 remaining solutions, to become the selected parents of prospective new offspring. Selection ensures fittest data is chosen to mate. The selection that only mates the fittest is an *elitist* and often leads to converging at local optima.
c. Crossover ensures best fit data (genes) are exchanged to yield a new and fitter pool. There are two crossover types (depends on encoding type used): (a) ***simple*** crossover for binary encoded pool, which allows single- or multi-point cross with all genes from single parent, and (b) ***arithmetic***

crossover allows new pool to be created by adding a data or individual's percentage to another.

d. Mutation alters chromosomes by changing its genes or its sequence, to ensure new pool converges to global minima (instead of local optima). Algorithm stops if optimal is found, or after number of runs if new pools are created (though computationally expensive), or when no better solution is found. Genes may change based on probability of mutation rate. Mutation improves the much needed diversity in reproduction and its algorithm is as thus:

Cultural GA (a variant) has belief space as thus: (a) Normative (has specific value ranges to which data is bound), (b) Domain (has data about task domain), (c) Temporal (has data about events' space is available), and (d) Spatial (has topographical data). In addition, an influence function mediates between its belief spaces and the pool – to ensure altered data conforms to the belief space. Thus, the data pool does not violate its belief space. This helps reduce number of possible individuals GA generates till an optimum is found (Reynolds, 2004).

C. Artificial Neural Network (ANN)

Inspired by human brain, an ANN consists of interconnected neurons that learn by example. Its neurons share data signals by adjusting its weight/bias (as connection strengths between synapses, axons and dendrites), which are summed, depending on the task by an activation function to yield an output that modulates its inputs and nonlinear feats exhibited as in Eq. 1 (Ojugo et al, 2012):

$$\emptyset = f(net) = f\sum_{i=1}^{m} Xi * Wij \qquad (1)$$

Thus, it translates into mathematics, principles of biological processing so as to generate predictive evolution outcomes of a task in the fastest time. Its derives outcomes from experience and recognizes behavior(s) as feats of interest from historic data, in order to suggest an optimal solution of high quality, irrespective of modification introduced to it by other methods in such a multi-agent space, which constantly affects the quality of any solution (Dawson and Wilby 2001). Encoded as a 3-layered unit of input, hidden and output, it is configured into ***feedforward*** and ***recurrent*** net based on task, data feat(s) to be approximated and connection requirements. Its learning allows a trained net to effectively classify patterns based on the employed learning rule into supervised, unsupervised and reinforcement (Ojugo et al, 2013).

The nature of fraud detection requires previous knowledge. Thus, we adopt the recurrent Jordan net to help us incorporate historic dataset feats of interest and previous output to be fed back as input into the model's hidden units to yield the next output. Its correlated weights are interconnected so that $W_{i,j}$ is weight between input and hidden layers, $W_{o}j$ is bias and x_i is diabetes input data. Its output is generated via tangent/sigmoid transfer function, which sums weighted input as in Eq. 2 and Eq. 3 (Minns, 1998). To resolve statistical dependency in the model's structure imposed by the data and heuristics adopted, our network used its ability to store earlier data generated from previous layer(s) as in Kuan and White (1994).

$$Z_{ij} = w_o j + \sum_{i=1}^{m} x_i * w_i j \qquad (2)$$

$$F(Z_{ij}) = \frac{2}{1 + e^{-2*Zij}} - 1 \qquad (3)$$

The Jordan network is more plausible and computationally more powerful than other models. Its back-propagation in time algorithm allows for advanced training/learning so that output at time *t* is used along with a new input to compute its output at time *t+1* in response to dynamism (Mandic and Chambers, 2001). It computes its output y^k, using the Tansig function that sums all inputs, receives target value of input training pattern and computes its error data, which is then sent back from its output to input nodes via error back-propagation to correct and update its weights and biases (c_j^k and c_o^k respectively) using mean square error. It then finds weights that approximate the target output with selected accuracy and modifies its weights by minimizing the error between ***target*** and ***computed*** outputs at end of each forward pass. If error is higher than selected value, process continues with reverse pass; else, stop training. Weights are updated till minimal error is found (Ojugo et al, 2013; Ursem et al, 2002; Guo and Xue, 2011).

Our Jordan net is constructed by modifying the multilayered feedforward with addition a ***context*** layer to help retain data between observations. At each move, new inputs are fed to the net. Previous contents of hidden layer is passed into context layer and later fed back into the hidden layer in the next time step. The context layer contains nothing initially. Output from the hidden layer after the first input will be same as if there is no context layer (Ojugo et al, 2013). Weights are computed same way for new connections to/fro the context layer from its hidden layer. Training aims at best fit data weights computed via Tansig function that assumes approximation influence of data points at its center so that the function decreases from its center (Perez and Marwala, 2011) with an Euclidean length (r_j) which yields distance between $y = (y_1,...,y_m)$ vector and its center ($w_{1j},...,w_{mj}$) given by Eq. 4 to Eq. 6 respectively as:

$$r_j = \left|\left| y - Y^j \right|\right| = \left\{ \sum_{i=1}^{m} (y_i - w_i j)^2 \right\} ½ \qquad (4)$$

The suitable transfer function is applied to r_j:

$$\emptyset(r_j) = \emptyset \left|\left| y - Y^j \right|\right| \qquad (5)$$

Finally, output k receives weighted combination as:

$$y^k = w_o + \sum_{j=1}^{n} \left(c_j^k * \emptyset(r_j) \right) = w_o + \sum_{j=1}^{n} \left(c_j^k * \emptyset ||y - Y^j|| \right) \qquad (6)$$

IV. Experimental Framework

The proposed system will resolve existing problems via:

a. Perform repetitive tasks without emotional defects
b. Embody the knowledge of human experts with the help of special software tools, manipulate data to solve problems and make decisions in that domain.
c. Processes are better formalized and defined on machines.
d. Automatic updating of the knowledgebase.
e. Processes are better formalized and defined on machines.

A. Material and Methods

Dataset contains 33,000 records of credit card transactions. Each record has 23-fields and our nondisclosure agreement prohibits us from revealing the details of the database schema as well as the contents of the data. But, it suffices to know that it is a common schema used by banks in Africa and Nigeria as part of the harmonization scheme. It contains information that

banks deem important for identifying fraudulent transactions. The dataset was already classified into fraudulent or non-fraudulent classes. From records, 38.2% are fraud transactions (emanating from product transaction, asset misappropriation, corruption and financial statement fraud). The sampled data is for a 24-month period. Note that the number of fraud records for each month varies, and the fraud percentages for each month are different from the actual real-world distribution.

B. Data Preprocessing

From original dataset, we prepared the data as suitable for use by the model by removing redundant fields. This helps to reduce the data size as well as speed up the learning heuristics, simplified the learning patterns and made the learned patterns more concise (as adapted from Stolfo et al, 2015). We also compared results of learning between our suitable data versus the original data, and saw no loss in accuracy.

Also, observed data had a skewed distribution of 34% fraud and 66% non-fraud). We adopt 34% fraud class distribution as complete dataset (training data for fraud is always insufficient and we are not expecting an artificially, higher fraud rate to accurately compute suspicion score for fraud patterns). We also must determine ***suspicion*** score for each rule generated by the rule-based model in conjunction with the GA operators to help optimize functions for our training data. And though there are no rules for splitting data, we split it as 50% training, 25% cross-validation and 25% testing for fraud distribution, which also yielded the best classifier for the model. Thus, we demonstrate that even with outliers and noise in dataset and with imprecision and ambiguities applied at its input, model effectively classifies transactions into its proper classes. Thus, GANN effectively scales up learning algorithms void of over-parameterization, over-training and over-fitting of data feats; while maintaining overall performance accuracy.

C. Proposed Model Framework

From Fig 1, the proposed model design employs these:

a. The rule-based system consists of *classifier* to propagate the IF-THEN rule values of selected data, enhanced them as predefined variables classification into intrusion types for fraud detection. The rule-based model is a production system with four (4) components: (i) ***rule set*** containing in each rule a pattern that determines applicability of the rule and corresponding operation to be performed if rule is applied, (b) ***knowledgebase*** (previous transaction set, classified into fraudulent and non-fraudulent using if-then rules as selected data feats), (c) ***control*** strategy specifies the order in which the rules are compared to those in the knowledgebase to find a match and it seeks also a way to resolve conflicts that arise when several rules are matched at the same time, and (d) a rule ***applier*** (Rich, Knight and Nair, 2009).

b. Jordan network provides a self-learning ability, optimized by the CGA's recombines and mutation of the rule-based dataset to train/test the system so that it autonomously classifies transaction into its class types.

c. Decision support – consists of predicted value output with automatic update of the knowledgebase, as transactions are diagnoses on its encounters of new data as in fig 1.

D. Genetic Algorithm Trained Neural Network (GANN)

GANN is initialized with the if-then rules as individuals, whose fitness is computed. 30-individuals are then selected via tournament method as new pool. It then determines mating individuals to yield solutions. We use a multi-point crossover and mutation to help the network to learn all the dynamic and non-linear feats in the dataset (as feats of interest). With mutation, suspicion score for each rule between 1-to-30 is then randomly generated using Gaussian distribution corresponding to crossover points (all genes are from single parent). As new parents contribute the rest to yield new individuals whose genetic makeup is a combination of both parents, mutation is also applied to yield 3-random genes. These further undergo mutation and are then allocated new random values that still conform to the belief space. These random values will range between 0 and 1, which yields the suspicion score for each transaction as generated for each account holder.

The number of mutation applied depends on how far CGA is progressed on the network (how fit is the fittest individual in the pool), which equals fitness of the fittest individual divided by 2. New individuals replace old with low fitness so as to create a new pool. Process continues until individual with a fitness value of 0.8 is found – indicating that the solution has been reached (Ojugo et al, 2013).

Initialization/selection via ANN ensures that first 3-beliefs are met; mutation ensures fourth belief is met. Its influence function influences how many mutations take place, and the knowledge of solution (how close its solution is) has direct impact on how algorithm is processed. Algorithm stops when best individual has fitness of 0 (Dawson and Wilby 2001).

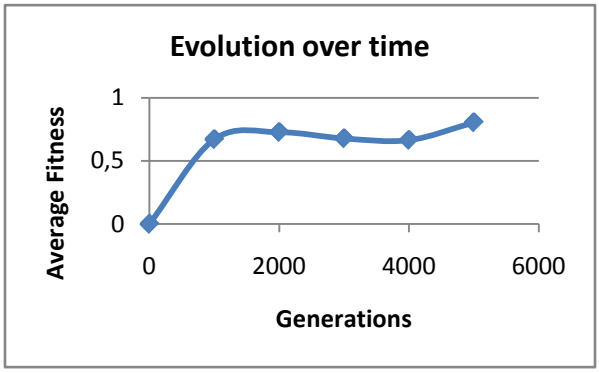

Fig. 1: Fitness Function Computation by Model

V. Result Findings And Discussion

For fraud detection, the performance rating of any detection mechanism is in its fraud catching rate and its false alarm rate. These are critical metrics such that a low fraud catching rate implies that a large number of fraudulent transactions will go through the system; Thus, costing the banks a lot of money (and the cost will eventually be passed to the consumers. Also, a high false alarm rate implies that large number of legitimate transactions will be blocked by the detection system. Thus, supervised intervention, monitoring and management will then be required to authorize transactions. This will frustrate many customers, while also adding operational costs (Stolfo et al, 2015). Note that the fraud catching rate is more important and critical than the false-alarm rate (true-negatives and false-

positives). Ideally, a cost function that takes into account true-negatives and false-positive rates, should be used to compare the classifiers. For lack of cost information from the banks, we rank our classifiers using first the fraud catching rate

Ojugo et al (2013) Performance is evaluated via computed values: mean square error and coefficient efficiency as thus:

Table 1. Model Convergence Performance Evaluation

Model	*MSE*	*COE*	*Fraud Catching Rate*	*False Alarm*
ANN	0.230	0.31	76%	24%
CGA	0.134	0.28	86%	14%
Rule-System			73%	27%
CGANN + Rule-based	0.313	0.219	91%	9%

After training and testing, compared to the models ANN, CGA and rule based system, the results are as follows: (a) ANN was run 24times and it took 223seconds to find solution after 98-iterations (best) and its fraud catching rate ranks at 76%. But, its demerit is that its solution is often trapped at local maxima, (b) GA was run 15-times to eradicate biasness and solution was found each time. It took 98seconds to find the solution after 123-iterations (best) and its fraud catching rate is 76%. Its convergence time depends on how close the initial population is to the solution as well as on the mutation applied to the individuals in the pool. Its demerit is that it seeks a global optima (in this case, a single rule that can be applied to all transactions. This would be appropriate if the transaction platforms are not considered as user are allowed to make transactions from various places – using varying devices that grants them access to their account at any point in time, and (c) CGANN with rule-based preprocessor hybrid was run 152times and its time varied between 29- and 245seconds to find solution after 102-iterations (best) and its fraud catching rate ranks at 91%. Its solution was made even closer using the fuzzy variable dataset (as a preprocessor).

Hybrids have proven to be intelligent modules to transform transaction with adaptive results that provides potential model for fraud detection. Its generated rule set has an accuracy of 92%, sensitivity of 91%, and failure analysis (true-negative and false-positive rate) of 14% respectively. However, the extracted rules are sound and agree with outcome of relevant fraud detection norms and studies.

A. Related Studies

Khasei et al (2012) adopted a feed-forward multi-perceptron network in their study. Its network was extended to represent complex dynamic patterns and cases to treats all data as new – so that previous data do not help to identify data feats chosen, even if such observed datasets exhibits temporal dependence. Consequently, this has practical implementation difficulty as large networks are not easily implemented. However, ***Jordan*** network overcomes such difficulty via its use of its internal feedbacks, which also makes it appropriately suitable for such dynamic, non-linear and complex tasks as its output unit is fed-back as input into its hidden unit with a time delay, so that its outputs at time $t-1$, is also input at time t.

Dheepa and Dhanapal (2009) also reviewed three methods to detect fraud by exploring the different views of same task to see what can be learned. They include: (a) clustering via data clustering of regions of parameter value, (b) Gaussian mix by a measure of the probability density of credit card user's past behavior, to compute probability of current behavior so as to detect anomalies from past behavior, and (c) Bayesian net is used to describe statistics of a particular user and the statistics of different fraud scenarios. He also suggested a combination of all three classifiers.

Stolfo et al (2015) employed meta-learning heuristics (ID3, Bayes, CART and RIPPER) in their study to properly classify transactions. In comparison, the Bayes, RIPPER and CART as base classifiers performed well in its classification of 80% fraud-catching and 19% false alarm result rates respectively.

B. Rationale for Statistical Technique

A major reason for using statistical-data analytics to tackle fraud owes from the fact that a lot of internal control systems have serious weaknesses such that for effective management, institutions must monitor and investigate every transaction that takes place and test it against established parameters, across apps and across systems, from dissimilar applications and data sources. Also, in implementing internal system, some controls are never even turned on. Thus, our rationale for adopting stochastic technique is based on Peter (2014) in which he compared convergence behavior against other machine learning techniques for task classification. In his comparison, decision trees takes 90iterations to converge; neural network approach takes 70iterations, clustering takes 40iterations; while, the hybrid model takes 30 iterations to converge. We also note that while trying to balance model's speed and greater accuracy of classification, more number of rule-set are generated and the knowledgebase consequently updated for optimality and greater functionality. Model trades off speed and accuracy for memory ***resource*** management.

VI. Conclusion and Recommendations

Hybrids are tedious and difficult to implement. Also, dataset accompanying the model must be appropriately encoded so that model can effectively exploit numeric data, which will in turn help the model efficiently, explore the problem space and yield optimal solution. Modelers must seek appropriate data feats and parameter selection alongside proper adjustment of weights/biases so as to avoid ***over-fitting***, ***over-training*** and ***over-parameterization*** of model. If data is properly encoded in a model's structured learning, it will help resolve the conflict imposed on its by the dataset and statistical dependencies of the varying heuristics used. Hybridization using CGA curbs rampant deviates along its output, imposed on model as agents in the space create and enforce their own behavioral rules on the dataset. This is achieved using the CGA belief space.

Models serve as educational tools to compile knowledge about a task, as new language to convey ideas as experts gain better insight to investigate input parameter(s) crucial to a task (Perez and Marwala, 2011), and its sensitivity analysis helps to reflect on theories of systems functioning. Simple model may not yield enough data; while complex model may not be fully understood. Detailed model helps us develop reasonably-applicable models even when not operationally applicable in a larger scale Their implementation should seek its feedback as more critical rather than seeking an accurate agreement with historic data. Since, a balance in the model's complexity will

help its being understood and its manageability, so that the model can be fully explored as seen here (Ojugo et al, 2012).

REFERENCES

[1] Bolton, R.J and Hand, D.J., (2002). *Statistical fraud detection: a review*, Statistical Science, 17(3), pp235-255.

[2] Chakraborty, R., (2010). *Soft computing and fuzzy logic*, Lecture notes, retrieved from http://www.myreaders.info/07_fuzzy_systems.pdf

[3] Coello, C., Pulido, G and Lechuga, M., (2004). *Handling multiple objectives with particle swarm optimization*, Proc. of Evolutionary Computing, 8, pp 256–279.

[4] Dawson, C and Wilby, R., (2001). *Comparison of neural networks in river flow forecasting*, J. of Hydrology and Earth Science, SRef-ID: 1607-7938/hess/2001-3-529.

[5] Delamaire, L and Abdou, H., (2009). *Credit card fraud and detection techniques: a review*, Banks and Bank Systems, 4(2), pp57

[6] Dheepa, V and Dhanapal, R., (2009). *Analysis of Credit Card Fraud Detection Methods*, Int. Journal of Recent Trends in Engineering, 2(3), pp126

[7] Guo, W.W and Xue, H., (2011) *An incorporative statistic and neural approach for crop yield modelling and forecasting*, Neural Computing and Applications, 21(1), pp109-117

[8] Heppner, H and Grenander, U., (1990). S*tochastic non-linear model for coordinated bird flocks*, In Krasner, S (Ed.), *The ubiquity of chaos* pp.233–238. Washington: AAAS.

[9] Kennedy, C and Porter, A., (2013). *Fraud detection and prevention*, [online]: retrieved July 2015 from www.moss-adam.com/fraud_reviews

[10] Khashei, M., Eftekhari, S and Parvizian, J (2012). *Diagnosing diabetes type-II using a soft intelligent binary classifier model*, Review of Bioinformatics and Biometrics, 1(1), pp9-23.

[11] Kuan, C and White, H., (1994). *Artificial neural network: econometric perspective*, Econometric Reviews, Vol.13, Pp.1-91 and Pp.139-143.

[12] Mandic, D and Chambers, J., (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Wiley & Sons: New York, pp56-90.

[13] Michalewicz, Z., (1998). *A survey of constraint handling techniques in Evolutionary computation methods*, www.dhpc.adelaide.edu.au

[14] Minns, A., (1998).*Artificial neural networks as sub-symbolic process descriptors*, published PhD Thesis, Balkema, Rotterdam, Netherlands

[15] Ojugo, A., Eboka, A., Okonta, E., Yoro, R and Aghware, F., (2012). *GA rule-based intrusion detection system*, Journal of Computing and Information Systems, 3(8), pp 1182 - 1194.

[16] Ojugo, A.A., Emudianughe, J., Yoro, R.E., Okonta, E.O and Eboka, A., (2013). *Hybrid neural network gravitational search algorithm for rainfall runoff modeling*, Progress in Intelligence Computing and Application, 2(1), doi: 10.4156/pica.vol2.issue1.2, pp22–33.

[17] Ojugo, A.A., Ben-Iwhiwhu, E., Kekeje, D.O., Yerokun, M.O and Iyawa, I.J.B., (2014). *Malware propagation on time varying network*,Int. J. Modern Edu. Comp. Sci., 8, pp25 – 33.

[18] Perez, M and Marwala, T., (2011). *Stochastic optimization approaches for solving Sudoku*, IEEE Transaction on Evol. Comp., pp.256–279.

[19] Peter, S., (2014). *An Analytical Study on Early Diagnosis and Classification of Diabetes Mellitus*, Bonfring International Journal of Data Mining, 4(2), pp7-13.

[20] Reynolds, R., (1994). *Introduction to cultural algorithms*, Transaction on Evolutionary Programming (IEEE), pp.131-139.

[21] Saleh Elmohamed, M.A, Fox. G and Coddington, P., (1998). *A comparison of annealing techniques for academic course scheduling*, Notes on Intelligence Computing, DHCP-045, pp 1-20. www.dhpc.adelaide.edu.au.

[22] Stolfo, S.J., Fan, D.W., Lee, W and Prodromidis, A.L., (2015). *Credit card fraud detection using meta learning: issues and initial results*, [online]: http://www.researchgate.net/publication/2282588

[23] Ursem, R., Krink, T., Jensen, M.and Michalewicz, Z., (2002). *Analysis and modeling of controls in dynamic systems*. IEEE Transaction on Memetic Systems and Evolutionary Computing, 6(4), pp.378-389

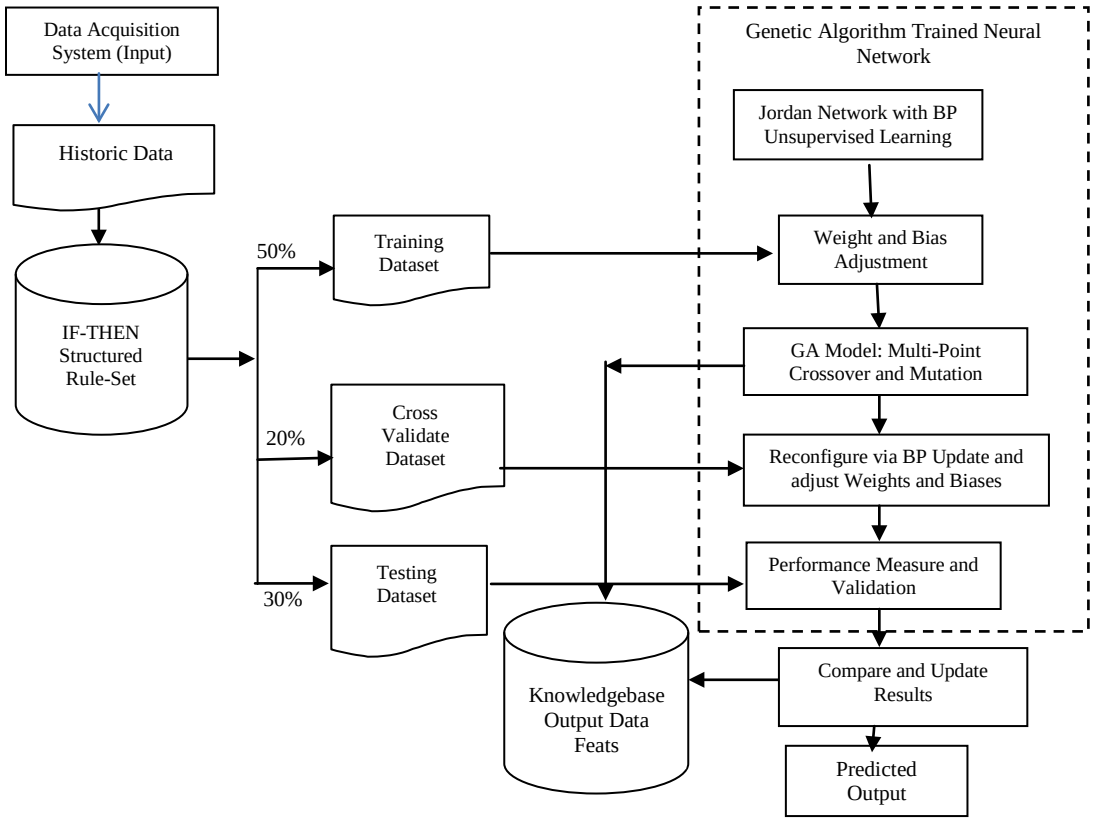

Fig. 2: Dataflow Diagram of the Hybrid Model Fuzzy Genetic Algorithm Trained Neural Network