Journal of Ubiquitous Systems & Pervasive Networks
Volume 15, No. 1 (2021) pp. 35-41

An Enhanced Deep Learning Model to Network Attack Detection, by using Parameter Tuning, Hidden Markov Model and Neural Network

Choukri Djellali*, Mehdi adda

Department of Mathematics, Computer Science and Engineering
University of Quebec At Rimouski
300 Allée des Ursulines, Rimouski, QC G5L 3A1
Rimouski, Canada,
Choukri_Djellali@uqar.ca, mehdi_adda@uqar.ca

Abstract

In recent years, Deep Learning has become a critical success factor for Machine Learning. In the present study, we introduced a Deep Learning model to network attack detection, by using Hidden Markov Model and Artificial Neural Networks.

We used a model aggregation technique to find a single consolidated Deep Learning model for better data fitting. The model selection technique is applied to optimize the bias-variance trade-off of the expected prediction. We demonstrate its ability to reduce the convergence, reach the optimal solution and obtain more cluttered decision boundaries.

Experimental studies conducted on attack detection indicate that our proposed model outperformed existing Deep Learning models and gives an enhanced generalization.

.

Keywords: *Deep Learning, Data Mining, HMM, Neural Network, Pattern Recognition, Model aggregation, Model selection, Network Security.*

1. Introduction

During recent years, Deep Learning (DL) has received tremendous attention due to its capability of searching complex decision boundary. This machine learning technique uses multiple processing layers to discover the underlying structures of data with multiple levels of abstraction.

It is defined as follows: *A sub-field of Machine Learning that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher-level concepts are defined from lower level ones, and the same lower level concepts can help to define many higher-level concepts* [1].

Formally, Deep Learning is an approximation of a target function ψ by a classifier ψ'' which is defined as follows:

$$\begin{cases} \psi : P \times C \text{ a } \{T, F\} \approx \psi'' : P \times C \text{ a } \{T, F\} \\ if\,(\psi(p_i, c_i) = T \Rightarrow p_i \in c_i \;\; else \;\; p_i \notin c_i \end{cases} \qquad (1)$$

The goal of DL is to select a function ψ that closely approximates a target function ψ'' by minimizing the generalization error defined by the following formula:

$$e = \arg\min_{\psi''}(\frac{1}{n}\sum_{i=1}^{i=n} f_L(\psi(p_i, c_i)) \qquad (2)$$

$\forall (p_i, c_i) \in S_n$

where $P \subset \Re^n$, $C \subset \Re^d$ and

$S_n = \{(p_1, c_1), (p_2, c_2), ..., (p_n, c_d)\},\;\; d \leq n.$

f_L : loss function.

DL has been increasingly used in several real-world applications such as Big Data [2], Medical diagnosis [3], Text Mining [4], Computational Biology [5], Neuroimaging [6], Cyber Security [7], and many others.

Recently, Artificial Neural Network (ANN) has been one of the most important sub-fields of Deep Learning.

* Corresponding author. Tel.: +1 418 723-1986
Fax: +1 418 724-1525; E-mail: Choukri_Djellali@uqar.ca

DOI: 10.5383/JUSPN.15.01.005

These models can be supervised-based or unsupervised-learning based. In supervised learning mechanism, the deep learning model learns from labeled training dataset. Unsupervised learning looks for correlated patterns in a data set to infer the hidden structure.

However, most DL models are sensitive to outliers, noise, presentation order, architecture configuration, and complex shapes.

On one hand, the model aggregation techniques are used to find a single consolidated Deep Learning model for better data fitting.

On the other hand, models selection is the most common technique in Machine Leaning, which is a meta-model or an averaging scheme designed to assess the learning stability and improve the recognition accuracy.

Based on these premises, we introduced a new Deep Learning scheme based on model aggregation and model selection.

The paper is organized as follows: In Section 2, we present the current state of the art in Deep Learning, our research questions and drawbacks of Deep Learning models. The conceptual architecture of our Deep Learning model is given in Section 3.

Before we conclude, we give in Section 4 an evaluation with a benchmarking model for Deep Learning.

Finally, a conclusion (Section 5) ends the paper with future works.

2. Literature review

In recent years, the use of Deep Learning has gained popularity in Data Mining. Several Neural Networks have been proposed to solve the problem of pattern recognition. They are designed to simulate the biological Neural Networks. The neural architecture is composed of many interconnected units usually known as artificial neurons. It is widely used for more complex tasks such as Categorization, Prediction, Clustering, Regression, and Summarization, etc. Among the most popular models in this category, we quote Self Organizing Map (SOM), Growing Neural Gas (GNG), Adaptive Resonance Theory (ART), Real-Time Recurrent Learning (RTRL), Gated Recurrent Units (GRUs), Boltzmann Machine, Learning Vector Quantization (LVQ), Deep Belief Networks (DBNs), Hopfield, Bidirectional Associative Memory (BAM), Growing Cell Structures (GCS), Recurrent Convolutional Neural Network (RCNN), etc [14].

This revolution that Deep Learning witnessing, has led to the appearance of several approaches.

(Yu et al., 2007) [8] introduced a graph-based consensus clustering (GCC) algorithm to find the underlying classes of the input vectors. Experimental results show that this model identified the number of clusters, discovered the clusters of input vectors, and outperformed the state-of-art models.

(Yang and Liu, 2019) [9] described an attack-resilient network connectivity model to facilitate the multi-UAV collaboration. To avoid the instability dilemma, this model used the conditional GAN with the least square objective loss function and Mean Square Error. The GAN paradigm is leveraged to characterize the adversary between a pair of UAVs and a malicious jammer. The baselines contain two GAN-based algorithms with three players and one non-GAN based game algorithm. Results demonstrate that the proposed model reduced the convergence, improved the connection latency, and enhanced the attack-resilience capability.

Recently, (Haddadpajouh et al., 2020) [10] introduced a multi-view Fuzzy consensus clustering model for malware threat attribution. To avoid a bias-variance dilemma, this model applied a fuzzy pattern tree and multi-modal fuzzy classifier. The consensus clustering technique is used to define an optimum distinction among the malicious actor's behaviour. The proposed model yielded 95.2% accuracy in pattern recognition task.

(Berti-Equille and Zhauniarovich, 2017) [11] presented an analytic pipeline to cluster and characterize attack campaigns into several profiles that exhibit similarities. To avoid multi-co linearity, this model applied data selection and normalization techniques. Ensemble learning combines multiple clustering techniques, such as K-Means, HDBSCAN, Self-Organizing Map (SOM), Hierarchical clustering (HCLUST), and EM (expectation maximisation). The posterior M5 and decision tree-based rules are discovered from consensus clustering.

(Sharma et al., 2017) [12] proposed a consensus framework for mitigation of zero-day attacks in IoT networks. During attack mitigation, the proposed model uses context behaviour of IoT devices, alert message protocol and data sharing protocol for reliable communication.

Experimental results showed that the proposed approach detected and eliminated the zero-day attacks in IoT network without compromising its performance.

(Rubio et al., 2020) [13] Introduced an approach for distributed detection of Advanced Persistent Threats or APTs. The evaluation based on testing set, showed that this model yielded good results, performed better than Opinion Dynamics based on consensus and presented an optimal traceability in a distributed setting.

(Choukri et al., 2020) [15] presented a new hybrid Deep Learning model based-Recommender System using Artificial Neural Network and Hidden Markov Model. The model aggregation technique is used to improve the robustness and accuracy of training. The model selection technique is applied to optimize the bias-variance tradeoff. Experiment results showed that our Deep Leaning model led to significant improvement over benchmarking model.

In most approaches, the learning performance is based on presentation order of training samples. Moreover, adapting the weight for each input vector is difficult in noise or outliers scenarios with huge training space.

To alleviate the problems mentioned above, we propose a conceptual scheme for Deep Learning from network attack data, by using parameter tuning, Hidden Markov Model and Neural Network.

In the next section, we introduce the Deep Learning architecture adopted in our approach.

3. Deep Learning architecture

Figure 1 shows schematically the functional process of our architecture.

The training algorithm learns from the training data and builds the relevant model. The data set acts as a source of knowledge in our approach.

Our Deep Learning model starts by feeding a machine learning algorithm from a large data set.

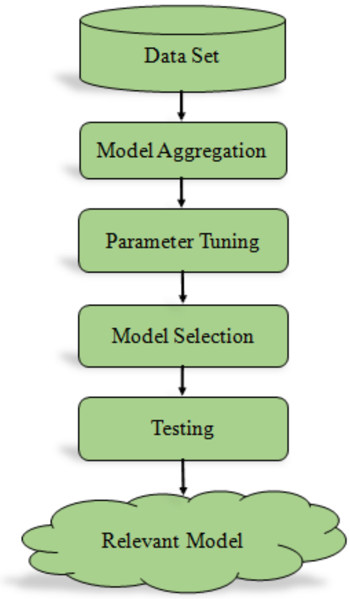


Fig. 1. Deep Learning architecture.

The Deep Learning model is based on model aggregation and model selection.

We used a Bootstrap aggregation scheme based on Hidden Markov Model (HMM) and Jordan Neural Network to classify patterns according to their contents [14].

This meta-modeling technique improves the stability and accuracy of pattern recognition algorithms.

To minimize the variance and bias of our model, we used an effective sampling technique based on k-fold Cross-Validation [18].

The data set is sampled into a training set and testing set. Of the k blocks, a single block is retained as the test data for performance evaluation, and the remaining (k – 1) blocks are used as training samples. The aggregate model created from a combination of aggregated models improves the stability and accuracy.

The obtained estimation of recognition accuracy from Cross-validation is not based on a selected model, but the average accuracies of trained models.

The neural architecture configuration needs careful tuning of parameters. We applied the typical initialization scheme to reduce the computation time and improve convergence speed of training.

The testing step is used to illustrate the robustness of our deep learning model.

Finally, the Deep Learning identifies the relevant model in the process of knowledge acquisition.

In the next section, we present our experimental studies of applying model aggregation, parameter tuning, and model selection. We first present the used data set for training, discuss the software configuration and then describe the parameters tuning for architecture configuration. We present the benchmarking models and describe the measures used for performance evaluation, and demonstrate the ability of our model to reach the optimal solution.

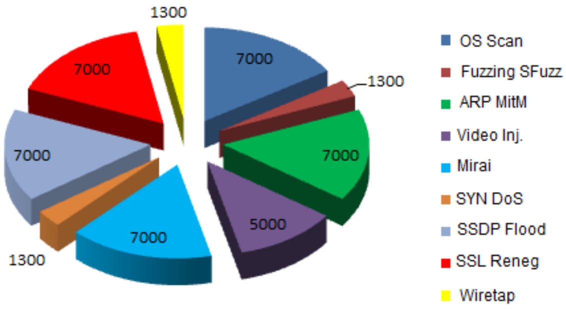
4. Experimental study

4.1. Configuration

Our proposed Deep Learning architecture has been implemented on Neon.1a Release (4.6.1) eclipse integrated development environment 64-bit and some library functions such as JDK 11.0.6 + Java EE, Java Matrix Package or JAMA[1], etc.

4.2. Data set

In our study, we used Kitsune Network Attack Dataset from Machine Learning Repository[2], which is the widely used data set for Network Attacks.



ıt figure

Fig. 2. Data set.

A frequency distribution illustrating the data is shown in Figure 2, with a mean of the sample size equal to 4877.77 and standard deviation equal to 2602.04. The number of patterns across categories is highly unbalanced.

We split our data into two sets, where 80% vs 20% for training and testing set respectively (no theoretical justification for this percentage).

4.3. Model aggregation

As mentioned in the previous section, we applied a bagging scheme based on Hidden Markov Model and Jordan Neural Network. The bootstrapping aggregation scheme is used to enhance the generalization ability of an ensemble of DL algorithms.

- *Hidden Markov Model or HMM*: is a statistical Markov model, in which the hidden states are modeled by a dynamic Bayesian network. The Baum-Welch algorithm is used for training, and Viterbi algorithm is applied to find the sequence of observed states from a given sequence [14].

[1] http://math.nist.gov/javanumerics/jama/

[2] https://archive.ics.uci.edu/ml/datasets/Kitsune+Network+Attack+Dataset

Formally, the Hidden Markov Model is a quintuplet, which is defined by

$$\lambda = (N, M, \mu, A, B)$$

Where,
λ : a parametric set.
N: the number of hidden states,

$$S = \{S_i\}_{\mathrm{i=1}}^{\mathrm{N}}$$

M: the number of distinct symbols observable by state,

$$o = \{o_k\}_{\mathrm{k=1}}^{\mathrm{M}}$$

$A = \{a_{ij}\}$: the transition matrix of hidden states,

$$a_{ij} = P\left(y_{t+1} = S_j / \, y_t = S_i, \lambda\right), 1 \leq i, j \leq \mathrm{N}$$

$B = \{bs_i(o_k)\}$: the probability matrix of observations

$$bs_i(o_k) = P(O_t = o_k / \, y_t = S_i, \lambda), i \leq \mathrm{N}, k \leq \mathrm{M}$$

$\mu = \{\mu_i\}_{i=1}^{N}$: the distribution of the initial state,

$$\mu_i = P(y_1 = S_i / \, \lambda)$$

- *Jordan Neural Network*: is a neural architecture that solves the categorization problem. It is three layered feed-forward Neural Network that learns an approximate function from learning samples. It learns the complex relationship between the variable predictors and target data, i.e., independent variables of the model.

As shown in Figure3, Jordan Neural Network consists of three layers of neurones, namely, the input layer, hidden layer, context layer, and output layer. The context layer represents the output feedback. All neurons in the hidden layer are fully connected to all neurons in the input layer and output layer.

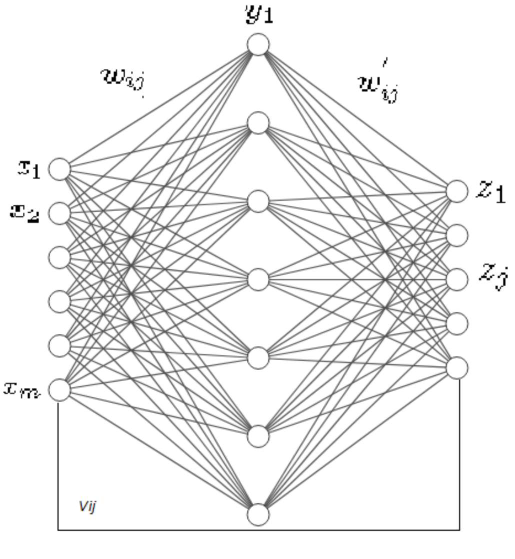


Fig. 3. Jordan Neural Network.

We used Batch Gradient Descent (or off-line) learning method to induce the relevant model. The cost function used to update the weights is defined as follow,

$$J(W) = \frac{1}{n}\sum_{i=1}^{n} J(W, x_i y_i \,)$$
$$W = W - \eta \nabla J(W)$$

The output of neuron *j* is defined as follows,

$$y_j(t) = f(a_j(t))$$
$$a_j(t) = \sum_{i=1}^{m} x_i(t) w_{ij}$$
$$z_j(t) = f(b_j(t))$$

$$b_j(t) = \sum_{i=1}^{m} y_i(t) w'_{ij} + \sum_{i=1}^{p} y_i(t-1) v_{ij}$$

The weights between hidden neurons and output neurons are adjusted as follows,

$$\Delta w'_{kj} = \eta \delta_k y_{\mathrm{j}}(\mathrm{t}) = \eta\left(o_k^D - z_{\mathrm{k}}(\mathrm{t})\right)\mathrm{f}'(a_{\mathrm{k}}(\mathrm{t}))y_{\mathrm{j}}(\mathrm{t})$$

The weight from the input layer to the hidden layer are updated by the following formula,

$$\Delta w_{ji} = \eta x_{\mathrm{i}}(\mathrm{t}) \delta_j = \eta x_{\mathrm{i}}(\mathrm{t}) \mathrm{f}'(a_{\mathrm{j}}(\mathrm{t})) = \sum_{k=1}^{c} w_{kj} \delta_{\mathrm{k}}$$

Context-Hidden weights,

$$\Delta v_{ji} = \eta y_{\mathrm{i}}(\mathrm{t}-1) \delta_j = \eta y_{\mathrm{i}}(\mathrm{t}-1) \mathrm{f}'(a_{\mathrm{j}}(\mathrm{t})) = \sum_{k=1}^{c} v_{kj} \delta_{\mathrm{k}}$$

The sensitivity of neuron *k* is defined as follow

$$\delta_k = \left(o_k^D - z_{\mathrm{k}}(\mathrm{t})\right)\mathrm{f}'(a_{\mathrm{k}}(\mathrm{t}))$$

4.4. Parameters tuning

Parameters tuning has a positive impact on Jordan training speed. We used the learning rate and activation function as two parameters to control the training speed.

The learning rate controls the training speed. When the learning rate increases, the Deep Learning achieves faster convergence. When the learning rate decrease, the Deep Learning model take a long time to converge or get stuck in suboptimal solution or local minimum solution.

We set the learning rate=0.001. Sigmoid is applied as an activation function for multiclass classification.

We used Mean Square Error (MSE) as a measure of how well the models fit data, which is the average squared difference between the desired outputs and current outputs. This criterion is defined as follows,

$$MSE = \frac{1}{n}\sum_{i=1}^{i=n}(o_i^c - o_i^d)^2$$

Where o_i^c stands for current output, o_i^d is for the desired output, and n is for the number of neurons in the output layer.

Figure 4 shows the learning curve on training data over a number of iterations. X-axis represents the number of iterations and Y-axis indicates the mean squared error.

At the start of the learning, the curve shows a high error indicating that the input training patterns are very spread out from the decision boundaries. At the end of the learning, the curve shows a low error, which means that the training is more reliable and the deep learning models reaches the optimal solution, and thus the model fits the input training patterns.

The recognition accuracy of Jordan based on typical initialization is equal to 96.91% after 53 iterations.

Compared to Jordan based on typical initialization, the recognition accuracy of Jordan Neural Network using Random initialization is equal to 95.71% with more training, i.e., the number of complete passes through the training patterns is equal to 155.

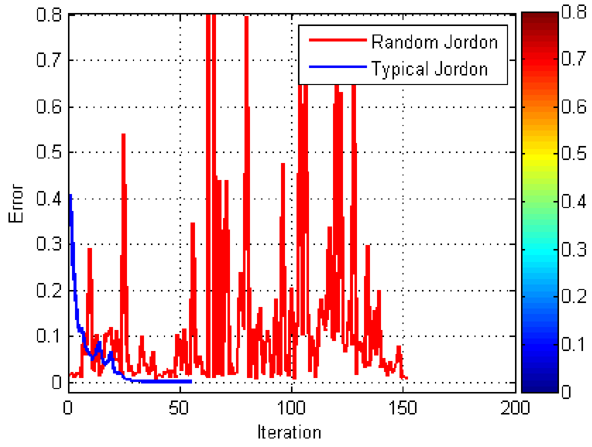


Fig. 4. Random initialization vs Typical initialization.

It is undoubtedly to see that Jordan Neural Network based on random initialization consumes more time.

Hence, the typical initialization scheme accelerates the convergence speed of training to reach the optimal solution. Moreover, the use of random initialisation adds unnecessary noise in the search space and affects the quality of learning, which negatively decreases the generalization.

In the light of the results, the typical initialization scheme is applied for training Jordon Neural Network.

In the next section, we present our experimental studies of applying model aggregation,

4.5 Learning

As mentioned in the previous section, we used a Bootstrap aggregation scheme based on Hidden Markov Model (HMM) and Jordan Neural Network.

The following figures show the learning curve on training data over a number of iterations. X-axis represents the number of iterations and Y-axis indicates the mean squared error.

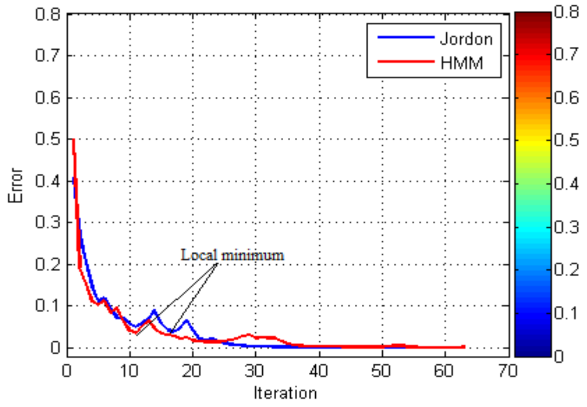


Fig. 5. Learning curve, HMM vs Jordan.

The training error does not decrease monotonically, it generally decreases, it can increase or oscillate. Hence, the DL algorithms search the complex shape of the decision boundaries and avoid local minimum during training.

The recognition accuracy of Jordan Neural Network is equal to 96.91% after 53 iterations.

Compared to Jordan Neural Network, Hidden Markov Model learns after 157 iterations with a recognition accuracy equal to 95.77% .

In the next section, we present an overview of the benchmarking models and describe in detail the metrics used to measure the Deep Learning performances.

4.6. Evaluation and baseline models

In this section, we review the evaluation measures used to test the robustness of our Deep Learning model. Four separate experimental studies were tested, which are used for comparison purpose. They correspond to Radial Basis Function, Bidirectional Associative Memory (BAM), ELMAN Neural Network, Fuzzy ART MAP and our Deep Learning model, namely DeepSecur, an abbreviation of Deep Learning Security .

This benchmark is designed to evaluate the learning performance in terms of recognition accuracies.

- *Radial Basis Function (RBF)*: is a Convolutional Neural Network (CNN) that contains an input layer, one hidden layer, and an output layer that computes the current outputs. These weights are adjusted by a supervised learning mechanism. The outputs are calculated using a non-linear RBF activation function. The feed-forward mechanism is used to adjust the weights of neurons.

- *Bidirectional Associative Memory (BAM)*: is a recurrent neural architecture that learns the long-term dependencies. The neural architecture is a hetero-associative memory that maps an input layer to an output layer. The Hebbian mechanism is used for synaptic weight learning.

- *ELMAN*: is a feed forward Neural Network architecture consisting of an input layer, a hidden layer, a context layer, and an output layer. The context layer is used to store the outputs of neurons in the hidden layer. The back-propagation algorithm is used to train the entries of the weight matrix.

- *Fuzzy ART MAP*: it is a Neural Network that has been successfully applied to design a stable model for plasticity-elasticity dilemma, i.e., incremental learning. The three-layered architecture contains two fuzzy ART Neural networks and an inter-ART module. The input layer of Neural Network is fully interconnected to its output layer, To categorize pattern with this neural architecture, many problems must be solved. Firstly, the neural network creates prototypes increasingly over time corresponding to the input patterns with high values. The prototypes with low values could never be accessed during the learning process. Therefore, the neural network prototypes are not accessible during the learning process, i.e., category proliferation [16,17]. Secondly, the random initialization reduces the convergence speed of clustering. Hence, the task of clustering with Fuzzy ART network requires a set of pre-processing operations before presenting the input vectors to the input layer.

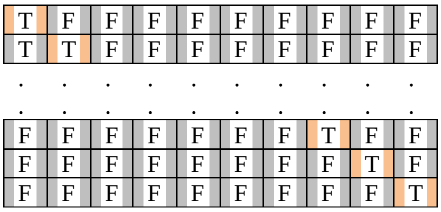
In order to overcome category proliferation obstacle, we used the complement coding of the input patterns. The complement coding allows a complete preservation of any information stored in the vector length, i.e., maintaining the amplitude of the vector and generating redundancy to distinguish the noisy variables, i.e., symmetric coding theory.

In order to obtain stable scoring results, we applied model selection techniques based on 10-fold Cross-Validation.

This sampling technique generates a diverse ensemble of classifiers by manipulating training and testing data.

By running repeated 10-fold Cross-Validation on training patterns, the aggregate estimation is defined as the average of the estimations obtained on each fold.

As shown in Figure 6, the data set is divided into training and test sets, with 9 folds (9F) to fit the model and 1 fold (T) to test its performance.



T	F	F	F	F	F	F	F	F	F
T	T	F	F	F	F	F	F	F	F
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
F	F	F	F	F	F	F	T	F	F
F	F	F	F	F	F	F	F	T	F
F	F	F	F	F	F	F	F	F	T

Fig. 6 10-fold Cross-validation

We used a priori knowledge about the data as a direct way to validate the results.

Classification Accuracy, $f - mesure$, $precision$, $recall$ measures are used for performance evaluation, which are calculated for all 10 folds.

$$CA_\mu = \frac{\sum_{i=1}^{c} tp_i + tn_i}{\sum_{i=1}^{c} tp_i + fp_i + tn_i + fn_i}$$

$$precision_\mu = \frac{\sum_{i=1}^{c} tp_i}{\sum_{i=1}^{c} tp_i + fp_i}$$

$$recall_\mu = \frac{\sum_{i=1}^{c} tp_i}{\sum_{i=1}^{c} tp_i + fn_i}$$

$f - mesure_\mu$ index weights average of the $precision_\mu$ and $recall_\mu$, i.e.,

$$f - mesure_\mu = 2 \times \frac{precision_\mu \times recall_\mu}{precision_\mu + recall_\mu}$$

where tp , fp and fn are true positive, false positive, and false negative, respectively.

The following table lists the performance of Deep Learning models in pattern recognition task.

. **Table 1.Accuracy metrics for different learning models**

Model/Meseaure	*precision*	*recall*	*f-mesure*	*CA*
RBF	81.91	79.26	78.98	95.99
DeepSecur	93.17	85.19	89.00	97.67
BAM	92.79	84.74	88.58	96.78
ELMAN	82.96	85.79	84.35	96.27
Fuzzy ART MAP	83.17	81.91	82.53	96.17

Our Deep Learning model (DeepSecur) has good performance, which provides a robust model for attacks detection. It maximizes the generalization ability and induces effectively the relevant model. In addition, our model escapes from local minimum to reach the global minimum. This feature was an ingredient key in the process of attack detection.

5. Conclusion

In this paper, we have introduced a Deep Learning model to network attack Detection, by using parameters tuning, model aggregation and model selection.

The Bootstrapping scheme based on 10-fold Cross-Validation is used to improve the learning stability and thus, to yield a consolidated model by combining multiple runs of Deep Learning algorithms.

We applied the model selection techniques to optimize the bias-variance tradeoff of the expected prediction.

The random initialization influences the learning efficiency and generates a noisy decision boundary.

Hence, the typical initialization scheme is applied to reduce the computation time, improve the convergence speed, and thus achieve the neighbourhood vicinity of the optimal solution.

Experiment results showed that our Deep Learning model led to significant improvement on Network attack detection.

In addition, our model escapes from local minimum to reach the global minimum, and induces effectively the relevant model.

6. Future Works

The purpose of our next work is to improve generalization performance, i.e., choose the best model that satisfies the trade-of bias-variance. A new algorithm can be designed based on Boosting and Ensemble learning techniques.

Acknowledgments

We would like to gratefully acknowledge the support of university of Quebec at Rimouski for funding of this research.

References

[1] Li Deng and Y Dong. Foundations and trends in signal processing. Signal Processing, 7:3–4, 2014. https://doi.org/10.1561/2000000039

[2] Ye Shi, Chin-Teng Lin, Yu-Cheng Chang Weiping Ding, Yuhui Shi, and Xin Yao. Consensus learning for distributed fuzzy neural network in big data environment. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020. https://doi.org/10.1109/TETCI.2020.2998919

[2] Ye Shi, Chin-Teng Lin, Yu-Cheng Chang Weiping Ding, Yuhui Shi, and Xin Yao. Consensus learning for distributed fuzzy neural network in big data environment. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020. https://doi.org/10.1109/TETCI.2020.2998919

[3] Jun Yang, Ziming Hou, Changjiang Wang, Hao Wang, and Hongbing Zhang. Gene expression profiles reveal key genes for early diagnosis and treatment of damantinomatous craniopharyngioma. Cancer gene therapy, 25(9):227–239, 2018. https://doi.org/10.1038/s41417-018-0015-4

[4] Peixin Chen, Wu Guo, Lirong Dai, and Zhenhua Ling. Pseudo-supervised approach for text clustering based on consensus analysis. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6184–6188. IEEE, 2018 https://doi.org/10.1109/ICASSP.2018.8462376

[5] Anna Hernandez Duran, ToddM Greco, Benjamin Vollmer, Ileana M Cristea, Kay Gr¨unewald, and Maya Topf. Protein interactions and consensus clustering analysis uncover insights into herpesvirus virion structure and function relationships. PLoS biology, 17(6):e3000316, 2019. https://doi.org/10.1371/journal.pbio.3000316

[6] Javier Rasero, Ibai Diez, Jesus M Cortes, Daniele Marinazzo, and Sebastiano Stramaglia. Connectome sorting by consensus clustering increases separability in group neuroimaging studies. Network Neuroscience, 3(2):325–343, 2019 https://doi.org/10.1162/netn_a_00074

[7] Mohamed Amine Ferrag, Messaoud Babaghayou, and Mehmet Akif Yazici. Cyber security for fog-based smart grid scada systems: Solutions and challenges. Journal of Information Security and applications, 52:102500, 2020. https://doi.org/10.1016/j.jisa.2020.102500

[8] Zhiwen Yu, Hau-San Wong, and Hongqiang Wang. Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics, 23(21):2888–2896, 2007. https://doi.org/10.1093/bioinformatics/btm463

[9] Bo Yang and Min Liu. Attack-resilient connectivity game for uav networks using generative adversarial learning. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pages 1743–1751, 2019.

[10] Hamed Haddadpajouh, Amin Azmoodeh, Ali Dehghantanha, and RezaMParizi. Mvfcc: A multi-view fuzzy consensus clustering model for malware threat attribution. IEEE Access, 8:139188–139198, 2020. https://doi.org/10.1109/ACCESS.2020.3012907

[11] Laure Berti- Equille and Yury Zhauniarovich. Profiling drdos attacks with data analytics pipeline. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 1983–1986, 2017. https://doi.org/10.1145/3132847.3133155

[12] Vishal Sharma, Kyungroul Lee, Soonhyun Kwon, Jiyoon Kim, Hyungjoon Park, Kangbin Yim, and Sun-Young Lee. A consensus framework for reliability and mitigation of zero-day attacks in iot. Security and Communication Networks, 2017, 2017. https://doi.org/10.1155/2017/4749085

[13] Juan E Rubio, Cristina Alcaraz, Ruben Rios, Rodrigo Roman, and Javier Lopez. Distributed detection of apts: Consensus vs. clustering. In European Symposium on Research in Computer Security, pages 174– 192. Springer, 2020. https://doi.org/10.1007/978-3-030-58951-6_9

[14] Duda, R.O., Hart, P.E., Stork, D.G.,. Pattern classification. John Wiley & Sons 2012.

[15] Djellali, Choukri and Mehdi Adda, A New Hybrid Deep Learning Model based-Recommender System using Artificial Neural Network and Hidden Markov Model, Procedia Computer Science, 175, 214-220, 2020, Elsevier. https://doi.org/10.1016/j.procs.2020.07.032

[16] Djellali, Choukri. A new conceptual model for dynamic text clustering Using unstructured text as a case. Proceedings of the 2014 International conference on Computer Science Software Engineering. ACM, 2014 p. 13.

[17] Fuzzy ART properties Huang, Juxin and Georgiopoulos, Michael and Heileman, Gregory L, Neural Networks, 8, 2, 203-213, 1995, Elsevier. https://doi.org/10.1016/0893-6080(94)00073-U

[18] block CV: An r package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. Valavi, Roozbeh and Elith, Jane and Lahoz-Monfort, Josée J and Guillera-Arroita, Gurutzeta, Methods in Ecology and Evolution, 10, 2,225-232, 2019,Wiley Online Library. https://doi.org/10.1111/2041-210X.13107