International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 01 | January - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

A Data Driven Model for Identifying Potential Poisoning Attacks on Machine Learning Models

Abid Husain Ansari[1], Prof. Virendra Verma[2]
Department of Computer Science and Engineering
PCST, Indore.[1,2]

Abstract: Due to the largeness of data to be analysed by online platforms, most of the big data models leverage machine learning in the backend. This has led to a new type of attack by users termed as adversarial machine learning in which the machine learning model used in the backend is targeted rather than the from end or the APIs. In conventional attacks, the first line of attack is the front end of the software. In this case, the machine learning model used in the backend is fed with bogus and/or deliberately falsified data to make it inactive. This is termed as adversarial machine learning attack or adversarial cyber-attack. It is extremely challenging to detect such attacks as there are no clear signs of attacks such as redirections, malicious code scripts, auto refresh tags etc. Instead, the data fed to the back-end machine learning model is targeted using adversarial data feeds. In this paper, a deep learning based model is used to detect such attacks. The proposed approach attains a precision value of 81% which is significantly higher compared to existing work.

Keywords: Big data, Dark Web, Socio-Technical data, Poisoning Attacks, Deep Learning, Precision.

I. INRRODUCTION

As the technological framework has shifted towards big data and machine learning, the types of attacks have also shifted in nature and have become much more complex in nature. A typical framework for any software utilizing machine learning is depicted in figure below.

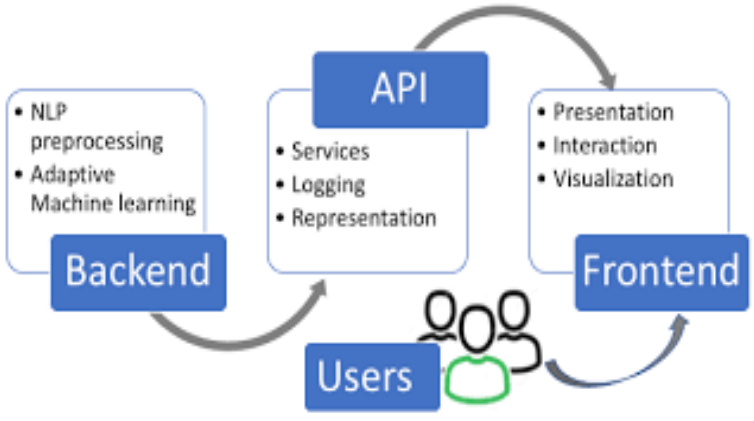


Fig.1 Machine Learning in Backend of Applications

With increasing refinement in the technological space, there lie high chances of sophisticated attacks to hinder the technological models. One such growing attack is the adversarial cyber-attack. An adversarial attack is a type of attack in which the attacker tries to feed wrong data into the training model dataset to train it to do something it

must not. In short, it poisons it to learn wrong information. Today it has become a lot more sophisticated that it becomes a difficult task to detect such attacks.

The attacks that were prevalent mainly targeted the availability or the integrity of the machine learning models. Feeding data that renders the entire system meaningless or useless has been one of its most prominent approaches. The consequences of such attacks are that the accuracy and performance drops drastically. Also, the purpose of the entire training concept gets falsified. There are some very huge threats that these attacks pose in the form of logic corruption, data modification, data injection and transfer learning. Henceforth there is a real need of detecting and preventing adversarial attack is need of the hour. With the increase in the machine learning approaches, the adversarial attacks have also increased at a fast pace. Now, the adversaries are using very high end tools to surpass the detection mechanisms. This is very legit concern for the artificial intelligence domain to safeguard itself from such threats.

II. PROPOSED MODEL TO DETECT ADVERSARIAL ATTACKS

The general function of social hacking is to gain access to restricted information or to a physical space without proper permission. Most often, social hacking attacks are achieved by impersonating an individual or group who is directly or indirectly known to the victims or by representing an individual or group in a position of authority [1]. In this case, it is assumed that the data on the profiles of hackers on dark web resources can render information about future trends and aspects of cyber attacks. The mathematical model of extraction of data from dark web forums is given below:

$$\boldsymbol{W(v_i, v_j) = \frac{1}{M}\sum_n \forall a, b : V(M, a)} \quad \textbf{(1)}$$

$$\boldsymbol{Or, W(v_i, v_j) = v_i^{(\beta\alpha^{(time, M_{k,b})} - (time, M_{k,a}))} + V(M_k, b)} \quad \textbf{(2)}$$

Here,
F is a dark web forum
W is the correlation between weights
n is the number of threads analysed
v is the number of users posting messages
M is the message number/index
k is the time index
(M_k, a) & $(M_k, b)$are the messages at time index k for distinct posts a and b in the same thread K.
α, β are constants with values between 0 and 1.
v_i, v_j are distinct messages

Another approach for estimating the similarity co-efficient or the distance among the messages is given mathematically as:
For two lists Γ^1 & Γ^2in the forum 'F', the similarity co-efficient or distance is computed as:

$$\boldsymbol{D^p(\Gamma^1, \Gamma^2) = \sum_{\forall\, i,j \in D(\Gamma^1\ \Gamma^2)} \widehat{D}^p_{i,j}\,(\Gamma^1\ \ \Gamma^2)} \quad \textbf{(3)}$$

Here,
Γ^1 & Γ^2 are two lists
D^p is the distance with a penalty p
$\widehat{D}^p_{i,j}$ takes up fuzzy values for different levels of similarity
(i,j) are the message pair

P is the optimistic penalty parameter
The above distance measure (Kendall's Measure) takes the relative ranking orders of any two elements in the union of two top k lists. Another measure is the absolute distance between the rankings of the same element in the union of two top k lists into consideration called the Spearman's distance measure given mathematically as:

$$F^{k+1}(\boldsymbol{\Gamma^1}, \boldsymbol{\Gamma^2}) = \sum_{i \in D_{r1} \cap D_{r2}} |\boldsymbol{\Gamma'_1} - \boldsymbol{\Gamma'_2}| \quad \textbf{(4)}$$

Here,

F represents the Spearman's distance

D_{ri} & D_{r2} represent the domains of $\boldsymbol{\Gamma^1}$ ***and*** $\boldsymbol{\Gamma^2}$

$\boldsymbol{\Gamma'_1}, \boldsymbol{\Gamma'_2}$ denoted the lists with/without entries in the original lists.

The mathematical representation of the adversarial-attack attack is given as:

Suppose the training vector be:

$$\boldsymbol{Training\ Data} = \boldsymbol{X(i)} \quad \textbf{(4)}$$

On Manipulation of the training vector using the poisoning vector stated by:

$$\boldsymbol{X_v} = \boldsymbol{V(i)} \quad \textbf{(5)}$$

The weights of the model are controlled by the training vector and the learning algorithm that is represented mathematically as:

$$\boldsymbol{w_k} = \boldsymbol{f(X_v, k, f_a)} \quad \textbf{(6)}$$

Here,
$X(i)$is the real training input
$V(i)$is the poisoning vector
k is the number of iteration
fa is the activation function
$w_k$is the weight for iteration k.

The poisoning attacks very surreptitiously try to hide behind the malicious dataset and try to deceive the machine learning classifier. When the data set itself is wrong and not the intended one, it trains the neural network to do something it shouldn't. Henceforth it is kind of a very tricky mechanism being developed as adversarial attacks. A very high end research is of paramount importance to able to mitigate the poisoning attacks that are prevalent. Off late there have been several cyber security practices that have come up to help the situation and deal with such sophisticated attacks. As these attacks are also of many types, a robust system is needed to thwart these types of attacks. The spamming of the data set is a popular kind of such an attack that tries to spam the training information with the adversary intended data. This can incur huge amounts of losses for the economy. Also, the wrongdoers will get a loophole to make use of to steal information and harm businesses. If there isn't any system to detect the adversarial attacks then malicious software and scripts can also be injected onto the dataset that can make the neural network behave erratically and work like a malware. Henceforth a strong poisoning attack detection scheme is important to safeguard the ML classifiers from improper datasets and malicious functions.

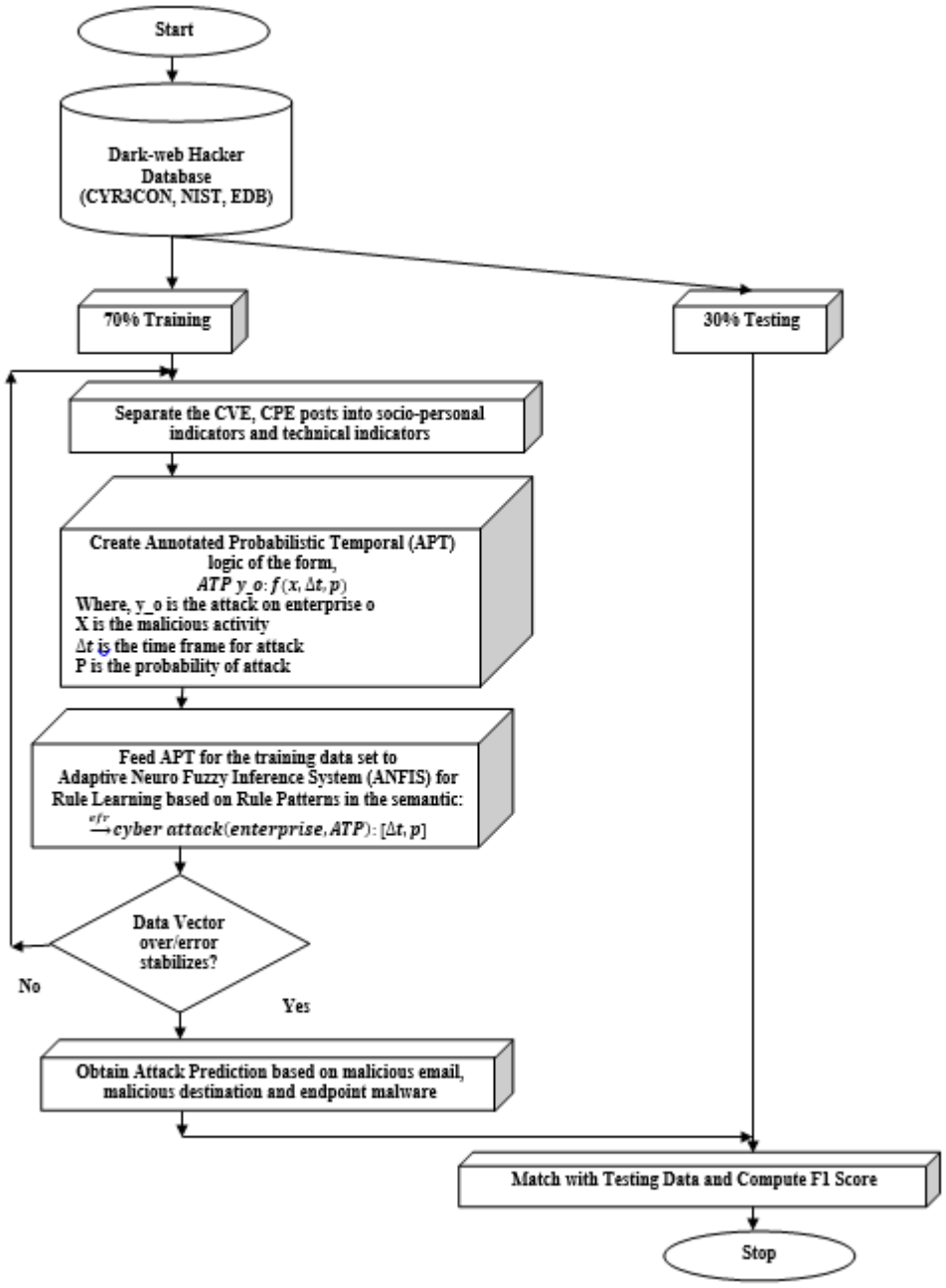


Fig.2 Proposed Flowchart

III MATHEMATICAL MODEL FOR PROPOSED WORK

In classical set, one can clearly understand that all sets have sharp boundary, along with their value of membership to being 0 & 1. Any fuzzy set can be expressed in following way:

$$\mu_x(A)\ \epsilon\ [0, 1] \qquad (7)$$

Where x is any set, A is any member of that set, $\mu_x(A)$ is membership function. Now similar to classical set theory fuzzy also exhibit almost similar type of operations. Let x & y are two fuzzy sets element

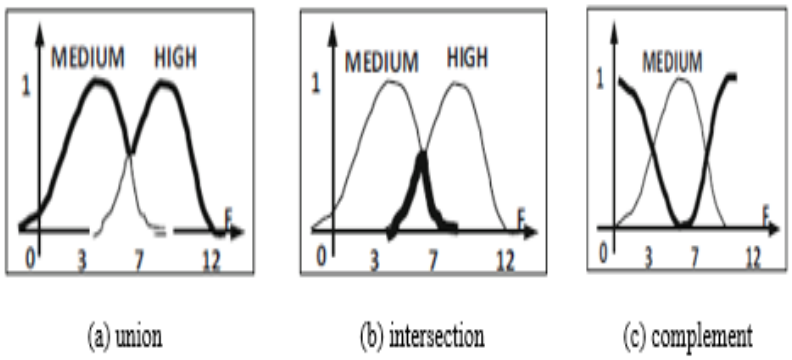
Union: $x\ U\ y = \mu_x(A)\ U\ \mu_y(A)$ **(8)**

$$\textbf{Union: } = \max\left(\mu_x(A), \mu_y(A)\right) \quad \textbf{(9)}$$

Intersection: $x \cap y = \mu_x(A) \cap \mu_y(A)$ **(10)**

$$\textbf{Intersection} = \min\left(\mu_x(A), \mu_y(A)\right) \quad \textbf{(11)}$$

Compliment: $x^c = F/x$ **(12)**



(a) union (b) intersection (c) complement

Fig.3 Fuzzy set operations

Any fuzzy logic controller performs three main tasks namely Fuzzification, Decision using control block and defuzzification. Post fis, the o/p generated is in the form of linguistic or vague form. In order to utilize in application, it need to be converted back to crisp o/p. Several methods are possible for defuzzification. some of them are discussed below.

Mean of maximum (MOM) method: in this method output value is equal to the average value of output with highest degree. Taking examples of AC again if the o/p FIS is “SLOW” then the crisp value from di-fuzzifier can be given by following equation.

MOM (SLOW) = $\frac{\sum x' \epsilon\, p^{x'}}{p} = \{x' | \mu_{SLOW}(x')$ **(13)**

Where.

p stands for set of o/p x with maximum value of MF for o/p SLOW.

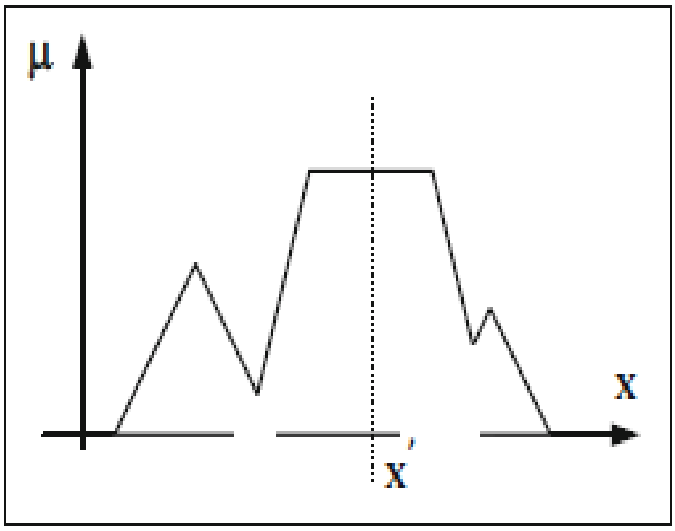


Fig.4 MOM method

The main issue with this technique is that it takes only maximum values in consideration. Hence if there are two membership functions of different shapes but same number of sets with similar highest degree will give same output.

Centre of Gravity (COG) method: As the name indicates, this method is based on the concept of centre of gravity of physics. In this method, unlike MOM the average value of all degree of belongingness for all values of output for a given MF. Taking same example of AC, for same categories of heater speed i.e. Slow, the equation for COG will be:

$$COG\ (SLOW) = \frac{\sum_X \mu_{SLOW}(x)\ X\ x}{\sum_X \mu_{SLOW}(x)} \qquad \textbf{(14)}$$

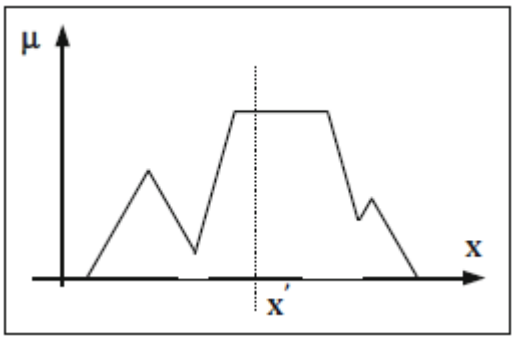


Fig.5 COG method

COG is far more popular than MOM because it considers complete area of MF. In many practical applications, COG is preferred for defuzzification. carrying output of each input set. In defuzzification process each subset of input are considered and rules are prepared and this are stored in look-up table.

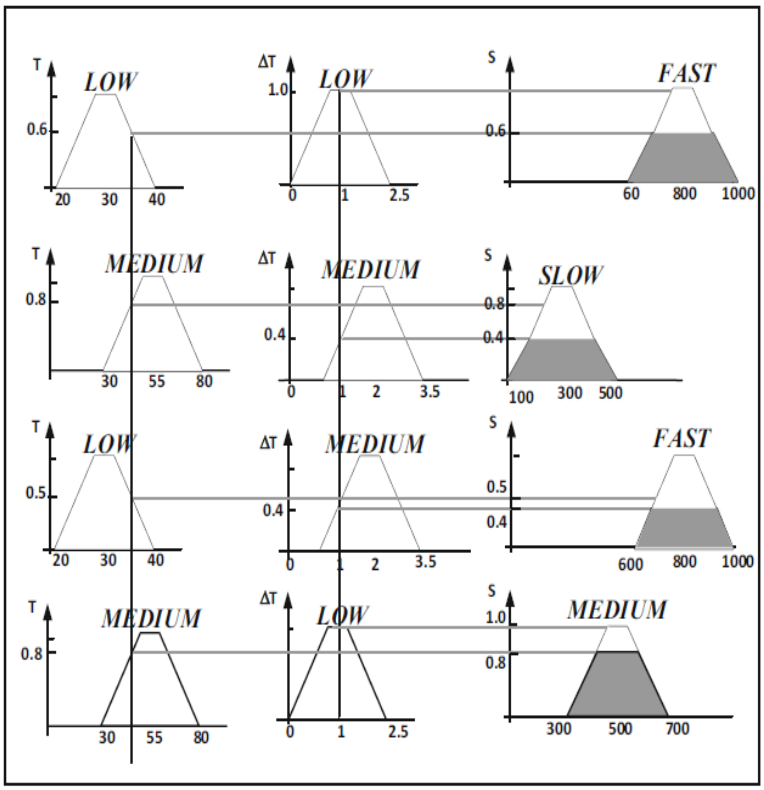


Fig.6 An illustration of the fuzzy output calculation

ANFIS stands for Adaptive Neuro-Fuzzy Inference System. The ANFIS controller combines the advantages of fuzzy controller as well as quick response and adaptability nature of ANN Fundamentally, ANFIS is about taking a fuzzy inference system (FIS) and tuning it with a back propagation algorithm based on some collection of input-output data. This allows your fuzzy systems to learn. A network structure facilitates the computation of the gradient vector for parameters in a fuzzy inference system. Once the gradient vector is obtained, a number of optimization routines is applied to reduce an error measure. This process is called learning by example in the neural network literature. Since ANFIS is much more complex than the fuzzy inference systems discussed so far, all the available fuzzy inference system options cannot be used. Specifically, ANFIS only supports Sugeno systems subject to the following constraints:

- First, order Sugeno-type systems.
- Single output derived by weighted average defuzzification.
- Unity weight for each rule.

An error occurs if your FIS matrix for ANFIS learning does not comply with these constraints. Moreover, ANFIS is highly specialized for speed and cannot accept all the customization options that basic fuzzy inference allows, that is, one cannot make own membership functions and defuzzification functions; that to make do with the ones provided. The overall performance metrics are mathematically defined as:

Accuracy: It is mathematically defined as:

$$Ac = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

Sensitivity: It is mathematically defined as:

$$Se = \frac{TP}{TP+FN} \quad (16)$$

Recall: It is mathematically defined as:

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

Precision: It is mathematically defined as:

$$Precisiosn = \frac{TP}{TP+FP} \quad (18)$$

F-Measure: It is mathematically defined as:

$$F - Measure = \frac{2.Precision.Recall}{Precision+Recall} \quad (19)$$

Here.
TP represents true positive
TN represents true negative
FP represents false positive
FN represents false negative

IV EXPERIMENTAL RESULTS

The system has been designed on MATLAB 2020a. The adversarial dataset has been collected for the dark web based entries. The choice of the tool has been made based on the ease of mathematical analysis and in-built mathematical functions. The obtained results have been presented subsequently.

Suspicious String Rates	Queueing delay	Response Delay of Server	Server Response Overhead	Colliding Token Percentage	Invalid tokens percentage	Redirection Tokensper	CPU Utilization Perce
0.225179552	0.130434783	0.00210615	0.691422856	0	0.03030303	0.024096386	0.602941176
0.420363329	0.057971014	0.002737995	0.025006252	0	0.151515152	0.024096386	0.808823529
0.562737643	0.072463768	0.00463353	0.133783446	0	0.060606061	0.036144578	0.926470588
0.394169835	0.173913043	0.00716091	0.403100775	0	0.060606061	0.156626506	0.794117647
0.029995775	0.089855072	0.011162595	0.263065766	0	0	0.228915663	0.720588235
0.269961977	0.15942029	0.0084246	0.461115279	0.105263158	0	0.036144578	0.867647059
0.029995775	0.252173913	0.007728011	0.24656164	0	0	0.084337349	0.838235294
0.19391635	0.420289855	0.00926706	0.736184046	0	0	0.084337349	0.852941176
0.269961977	0.101449275	0.010741365	0.140785196	0.052631579	0	0.13253012	0.588235294
0.108998733	0.333333333	0.006139427	0.417604401	0	0	0.168674699	0.735294118
0.483734685	0.113043478	0.005370682	0.500375094	0	0	0.072289157	0.882352941
0.19391635	0.727536232	0.011162595	0.392848212	0	0	0.120481928	0.808823529
0.19391635	0.333333333	0.004422915	0.2028007	0	0.03030303	0.120481928	0.823529412
0.225179552	0.275362319	0.004991575	0.478619655	0	0.090909091	0.108433735	0.823529412
0.483734685	0.191304348	0.003053917	0.600150038	0	0.060606061	0.036144578	0.926470588
0.225179552	0.086956522	0.005370682	0.134783696	0	0.03030303	0.096385542	0.897058824
0.674271229	0.223188406	0.007476832	0.31207802	0	0.090909091	0.060240964	0.823529412
0.394169835	0.420289855	0.006002527	0.297074269	0	0.03030303	0.084337349	0.882352941
0.517955218	0.593478261	0.0042123	0.469867467	0	0	0.144578313	0.823529412
0.690747782	0.195652174	0.007476832	0.620655164	0	0.060606061	0.096385542	0.823529412
0.225179552	0.115942029	0.004001685	0.135033758	0	0	0.036144578	0.573529412
0.029995775	0.086956522	0.008319292	0.057764441	0	0	0.060240964	0.617647059
0.151246303	0.15942029	0.002611626	0.329332333	0	0	0.120481928	0.897058824
0.225179552	0.68115942	0.004317607	0.370092523	0	0	0.048192771	0.676470588
0.09252218	0.275362319	0.003053917	0.243810953	0	0.03030303	0.096385542	0.882352941
0.311364597	0.275362319	0.00547599	0.270567642	0	0.03030303	0.120481928	0.823529412
0.378115758	0.391304348	0.005139006	0.184046012	0	0.03030303	0.096385542	0.897058824

Fig.7 Raw Data

The parameters used for identifying possible attacks are:

1) Suspicious String Rates
2) Queuing Delay
3) Response Delay of the server
4) Server Response Overhead
5) Colliding Token Percentage
6) Invalid Tokens Percentage
7) Redirections Token Parentage
8) CPU utilization percentage

The designed ANFIS model is shown in the figure. It is obtained by selecting the structure in the ANFIS model. The basic sections of the model are:
1) The input.
2) Input membership functions.
3) Rule generator.
4) Output membership functions.
5) Output.

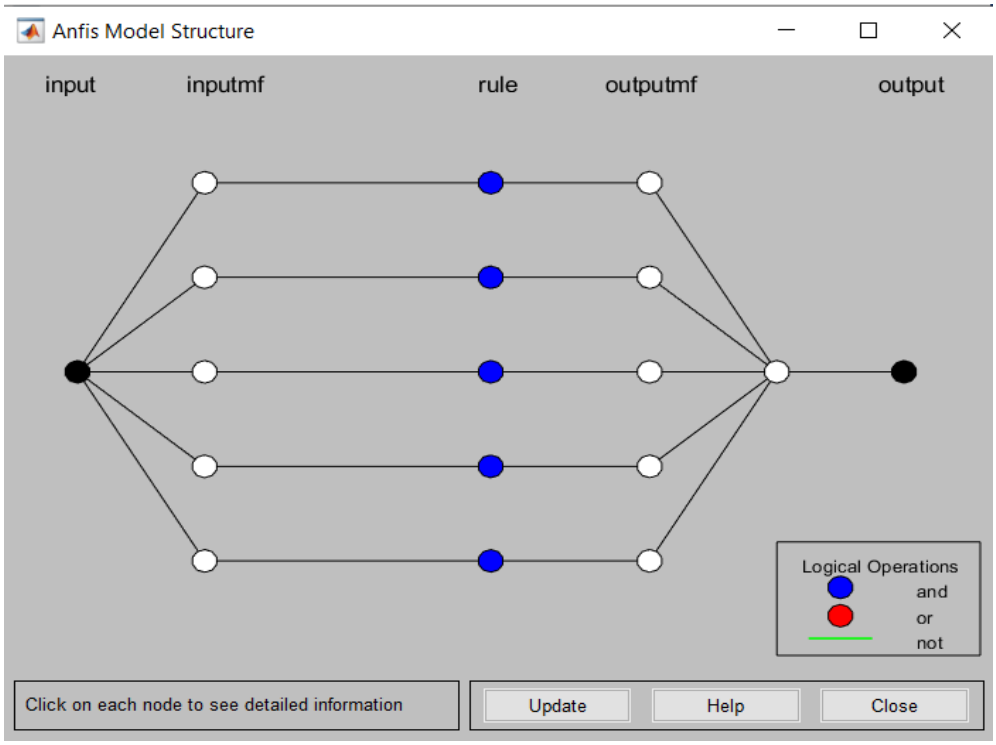


Fig.8 Structure of the ANFIS Model

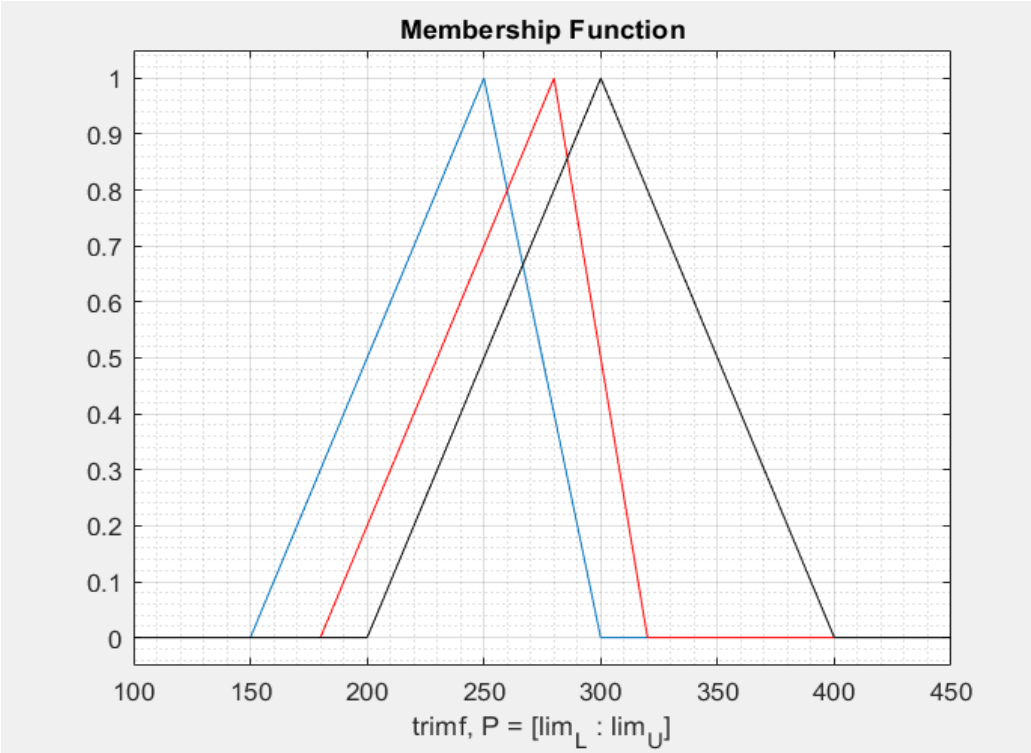


Fig.9 Membership Functions of Fuzzy System

The membership function used in the work have been depicted in figure 9. The memberships are shown to follow the triangular memberships with the following attributes:

1) $lim_L < value < lim_U$
2) The output y-axis range is in between 0 and 1.
3) There are overlapping boundaries among the membership functions.

A three tier membership is used which are:

1) Low
2) Moderate
3) High

The total number of samples used are 85,383. 30% of the data i.e. 25,615 samples have been tested for the possibility of attacks.

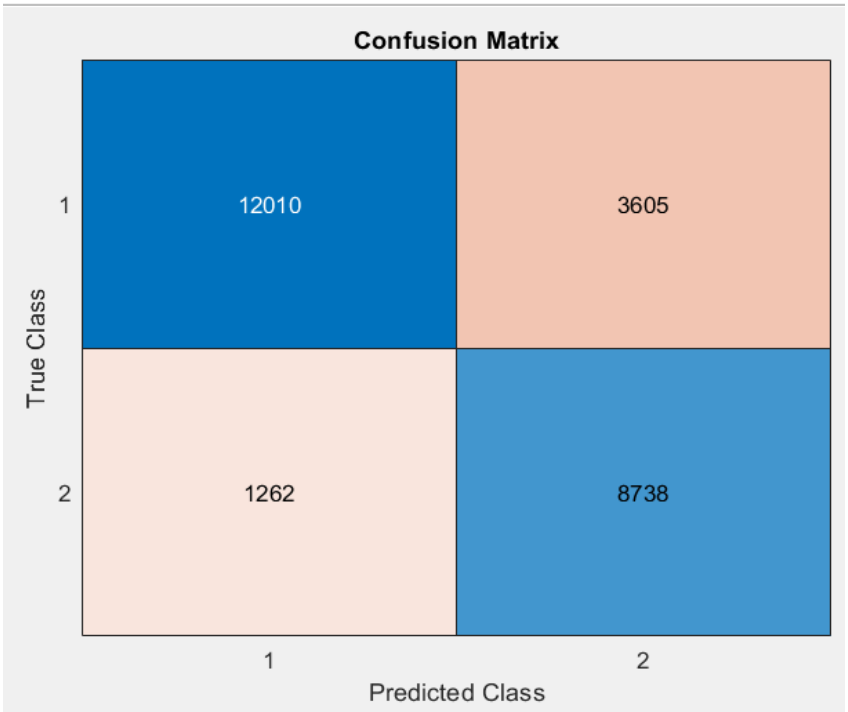


Fig.10 Confusion Matrix

The computation of the precision and accuracy can be done based on the values of TP, TN, FP and FN.

Table 1. TP, TN, FP and FN values.

S.No.	TP	TN	FP	FN
1.	12010	8738	3605	1262

The accuracy is computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 80.99\%$$

The precision is computed as:

$Precision = \frac{TP}{TP+FP}$ =76.91%

The previous work [1] attains a maximum precision value of 72.9%. Thus the proposed work outperforms the existing work in terms of classification accuracy.

Conclusion: It can be concluded from the previous discussions that adversarial machine learning based cyber-attacks have become a major challenge as the detection is not straightforward. With the advancement in technology, there has also been increase in cyber attacks and security breaches. Using machine learning models to predict security threats has many open research fields including predicting whether vulnerability would be exploited based on Dark Web sources. This paper presents an ANFIS based algorithm for detecting possible adversarial attacks in advance and hence can be thwarted. The results show that the proposed work attains higher classification accuracy and precision compared to existing techniques.

References

[1] A Paudice, L Muñoz-González, A Gyorgy, EC Lupu, “Detection of adversarial training examples in poisoning attacks through anomaly detection”, IEEE Transactions on Dependable and Secure Computing, 2023, pp.1-10.

[2] G. Li, J. Wu, S. Li, W. Yang and C. Li, "Multitentacle Federated Learning Over Software-Defined Industrial Internet of Things Against Adaptive Poisoning Attacks," in IEEE Transactions on Industrial Informatics, 2022, vol. 19, no. 2, pp. 1260-1269

[3]W. Jiang, H. Li, S. Liu, X. Luo and R. Lu, "Poisoning and Evasion Attacks Against Deep Learning Algorithms in Autonomous Vehicles," in IEEE Transactions on Vehicular Technology, vol. 69, no. 4, pp. 4439-4449, April 2020

[4]E. Marin, M. Almukaynizi and P. Shakarian, "Reasoning About Future Cyber-Attacks Through Socio-Technical Hacking Information," IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 157-164. [5] L Muñoz-González, B Biggio, A Demontis, “Towards adversarial of deep learning algorithms with back-gradient optimization”, ACM 2017

[6] S Shen, S Tople, P Saxena,” Auror defending against adversarial attacks in collaborative deep learning systems”, ACM 2016

[7] N Papernot, P McDaniel, S Jha,” The limitations of deep learning in adversarial settings”, IEEE 2016

[8] P Russu, A Demontis, B Biggio, G Fumera,” Secure kernel machines against evasion attacks”, ACM 2016

[9] F Zhang, PPK Chan, B Biggio,” Adversarial feature selection against evasion attacks”, IEEE 2015

[10] H Xiao, B Biggio, B Nelson, H Xiao,” Support vector machines under adversarial label contamination”, Elsevier 2015

[11] B Biggio, SR Bulò, I Pillai, M Mura,” Adversarial complete-linkage hierarchical clustering”, Springer 2014

[12] M Mozaffari-Kermani, S Sur-Kolay,” Systematic adversarial attacks on and defenses for machine learning in healthcare”, IEEE 2014

[13] X Lin, PPK Chan,” Causative attack to incremental support vector machine”, IEEE 2014

[14] N Hoque, MH Bhuyan, RC Baishya, “Network attacks: Taxonomy, tools and systems”, Elsevier 2014

[15] B Biggio, I Corona, D Maiorca, B Nelson, "Evasion attacks against machine learning at test time", Springer 2013

[16] B Biggio, B Nelson, P Laskov, "Adversarial attacks against support vector machines", Axirv 2012

[17] B Biggio, I Corona, G Fumera, G Giacinto, "Bagging classifiers for fighting adversarial attacks in adversarial classification tasks", Springer 2011

[18] J Long, W Zhao, F Zhu, Z Cai, "Active Learning to Defend Adversarial Attack Against Semi-Supervised Intrusion Detection Classifier", World Scientific Journal 2011