International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Detection of Web based-Attacks Using Artificial Intelligence

Y Rajesh[1], Ch. SivaKrishna[2], B. Jaswanth[3], J. SriHarsha[4]
[1]Assistant Professor, Department of Computer Science and Engineering, ALIET, Vijayawada, India
[2-4]UG Scholar, Department of Computer Science and Engineering, ALIET, Vijayawada, India

ARTICLEINFO

Article History:
Accepted: 01 April 2023
Published: 20 April 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
403-417

ABSTRACT

The cyber-physical systems (cps) have improved significantly in many dynamic applications. Yet, these systems are seriously threatened by cyberattacks. Cyber-attacks, in contrast to flaws that result from errors in cyber-physical systems, are premeditated and covert. Some of these assaults, also known as deception attacks, manipulate certain cyber components or introduce phoney data from sensors or controllers into the system to damage data or introduce misleading information. Inability to recognise these attacks if the system is not aware of them could cause performance issues or even total system failure. To identify these attacks in these systems, algorithms must be altered. It should be mentioned that since the data generated by these systems is produced in such vast quantities, with such a broad diversity, and at such a quick rate, the application of machine learning algorithms is needed to make it easier to analyse, appraise, and uncover hidden patterns in the data. The CPS is modelled in this study as a network of moving agents, one of which acts as the leader and the rest of which are submissive to it. The approach proposed in this study makes advantage of the deep neural network's structural features for the detection stage, which ought to alert the system to the attack's presence in its first stages. In a leader-follower mechanism, to isolate the problematic agent, The application of resilient control methods in a network has been researched. The control system in the previously discussed control approach use a deep neural network to identify an attack before isolating the attacking agent and then isolating the misbehaving agent using a reputation mechanism. Experimental investigations have shown that deep learning algorithms may identify assaults more precisely than traditional techniques, as well as simplify, be proactive, cost-effective, and greatly enhance cyber security.

Keywords: CPS, Cyber-Attacks, Cyber-Physical Systems

I. INTRODUCTION

The creation of cyber-physical systems as a result of recent technological advancements has greatly improved many dynamic applications because of their greater processing and communication capabilities as well as the integration of physical and cyber components. Yet, the downside of this progress is that it makes us more susceptible to cyberattacks. Cyber-physical systems consist of embedded computers and

logical components that connect through networks like the Internet of Things (IoT). In particular, these systems involve people who have received training in interacting with analogue, digital, and physical technology as well as both tangible and virtual components.Any system that has the ability to exchange goods between humans, physical components, and cyber components is considered a cyber-physical system. The integration of the physical component has made the security of cyber-physical systems more crucial than ever.

The system's physical components, such as the sensors that gather data from the outside world, are vulnerable to assault and might get false data as a result. A cyber-physical system's physical presence in the environment of several sensors that rapidly and effectively gather a lot of data from various sources is one of the primary problems with such a system. One of the primary obstacles will be integrating the sensors with the proper computations and performing an analysis of the data that is received.As a result, the ability to control the system, do computations, and coordinate the transmission of various sensors is one of a cyber-physical system's most crucial features.A fundamental problem with these systems is the security of cyber-physical systems to identify cyber-attacks. It should be highlighted that because they are unpredictable, it is hard to define cyberattacks systematically. Denial of service (DoS) attacks and deception assaults are the two main categories of cyberattacks in cyber-physical systems. Denial of service attacks stop network nodes and channels from talking to one another. Nevertheless, deception attacks that include modifying system parts like sensors or controllers may skew data or bring false information into the system, resulting in inappropriate behaviour.By monitoring the system, the system is able to find these assaults. Nevertheless, these attacks are known as stealthy deception attacks, and other regularly used measures to avoid them would be ineffectual if the attacker could organise a big attack to conceal himself from discovery. For a timely response to attackers, it is important to be aware of attacks as they happen. Or, to put it another way, information about the assault is necessary for the security system to recognise it and prevent it. Using security analytics to look for covert trends and deceitful tactics might improve cyber defence.

II. LETERATURE SURVEY

Weiyi Liu, Kwon, Cheolhyeon, and Inseok Hwang. "Security study against stealthy deception attacks for cyber-physical systems." American Control Conference, 2013, IEEE, 3344–3349

Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “ Awesome App ” . One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drastically Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one

to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drastically Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically

users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve

stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically

users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to · ve stars, with one star denoting a high sense of dissatisfaction with the app and · ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, o · en the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; "Awesome App". One would reasonably expect the rating for this review to be · ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de · ated (or in · ated) overall average rating of an app which can a · ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. · is is especially signi · cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall

A networked control system examines the security issue in the state estimation problem (NCS). The remote estimator and sensors of the NCS are connected by communication paths that might be hacked by hostile parties. We take into account attacks that introduce fake data. The aim of this work is to identify the so-called insecurity criteria that make the estimate system unsafe in the sense that even when malicious assaults are successful in evading the anomaly detector, they may still lead to an infinite number of estimating mistakes.The compromising of all communication channels, for example, results in the creation of a new required and sufficient condition for the vulnerability. The development of assaults that might endanger the estimating system is also advised to use a specific method. A system security approach for the insecure system is also provided, whereby just a small number of communication channels (as opposed to all of them) need to be protected from attacks such as the introduction of fake data. The usefulness of the suggested criteria and approaches in solving the secure estimate issue for a flying vehicle is demonstrated using a simulated scenario.

The authors are Miroslav Pajic, James Weimer, Nicola Bezzo, Oleg Sokolsky, George J. Pappas, and Insup Lee. Attack-resilient state estimators are the main subject of the paper "Design and implementation of attack-resilient cyberphysical systems." 37, no. 2 (2017): 66–81 in IEEE Control Systems Magazine.

Control system-related security-related incidents have significantly increased in frequency in recent years. They include well-known assaults on essential infrastructure (like the Maroochy Water spill) and complex industrial systems (like the German Steel Mill hack and the Stuxnet virus attack on an industrial supervisory control and data collection system). Even high-assurance military technology is vulnerable to assaults, as the widely reported downing of the US drone RQ-170 Sentinel demonstrated.With cyberphysical systems (CPSs), which intimately integrate computing and communication substrates with sensing and actuation components, these mistakes have considerably increased the necessity for security. Yet the next generation of networked, embedded, and safety-critical control systems is complex and diverse, challenging the conventional design approaches where security is sometimes considered an afterthought.

Sheng, Long, Ya-Jun Pan, and Xiang Gong. "Consensus formation control for a class of networked multiple mobile robot systems." Journal of Control Science and Engineering 2012 (2012).

Environments and requirements. Tracking and understanding changes in modules and relationships in a software project is difficult, but even more so when the software goes through several types of changes. The typical complexity and size of software also makes it harder to grasp software evolution patterns. In this paper, we present an interactive matrix-based visualization technique that, combined with animation, depicts how software designs evolve. For example, it shows which new modules and couplings are added and removed over time. Our generic visualization supports dynamic and weighted digraphs and is applied in the context of software evolution. Analyzing source code changes is important to determine the software's structural organization and identify quality issues over time. To demonstrate our approach, we explore open-source repositories and discuss some of our findings regarding these evolving software designs

The availability of embedded computational resources in autonomous robotic vehicles has increased the operational efficacy of cooperative robotic systems in both civilian and military applications. Cooperative collaboration is more effective and capable of completing tasks as compared to autonomous robotic vehicles that carry out activities independently. The control of many robots, platooning of vehicles for urban transportation, autonomous underwater vehicles, and formation of planes for military operations are just a few of the many uses for multirobotic vehicle systems.The primary goal is to investigate collective behaviour in multirobot systems. Cooperative group members work for a single goal and represent the interests of the whole group. The success of group collaboration depends on how well group members coordinate their efforts. Each group member may cooperate to encourage cooperative behaviour via either local or global coordination. Like schooling fish, humans only respond to their immediate neighbours when coordinating locally.

Zeng, Wente, and Mo-Yuen Chow. "Resilient distributed control in the presence of misbehaving agents in networked control systems." IEEE transactions on cybernetics 44, no. 11 (2014): 2038-2049.

The difficulty of getting all the agents in networked control systems (NCS) to agree in the face of misbehaving agents is examined in this article. A reputation-based robust distributed control mechanism is originally proposed for the leader-follower consensus network. The recommended approach divides the control process into the four resilience mechanism phases of detection, mitigation, identification, and update.In order to detect and isolate the misbehaving agents and even lessen their influence on the system, each agent only utilises information about nearby neighbours and one-hop neighbours throughout each phase. Then, we incorporate two recovery mechanisms (rollback and excitation recovery) into the current architecture to guarantee the accurate convergence of the proposed algorithm to the leaderless consensus network. Using case studies in wireless sensor networks and multirobot formation control, the usefulness of the suggested strategy is illustrated.

Sun, Hongtao, Chen Peng, Taicheng Yang, Hao Zhang, and Wangli He. "Resilient control of networked control systems with stochastic denial of service attacks." Neurocomputing 270 (2017): 170-177.

A Markov process is used to characterise packet dropout since it is based on the competition between attack and defence strategies. Four theorems are established for system stability analysis and controller design, and an NCS is then simulated using a Markovian jump linear system under these game results. To show how these theorems might be used, a numerical example is supplied at the end. Networked

control systems (NCSs) have been more popular over the last few years. It is increasingly commonplace to employ NCSs in industrial operations, electrical power networks, intelligent transportation, and other sectors. Network has been increasingly important with the growth of NCSs.

III. Proposed System

The issue of misdiagnosis in predicting whether a cyberattack will occur has been addressed by a number of machine learning algorithms, but none of them has been able to fully resolve it. In past research that have offered models for evaluating this performance categorization, the variability and magnitude of the data are frequently overlooked. We suggest using the Support Vector, Decision Tree, Random Forest, Extra Tree Classifier, Ad Boost, and Neural Network Classifier techniques to categorise data.

Advantages

- Highest accuracy
- Reduces time complexity.

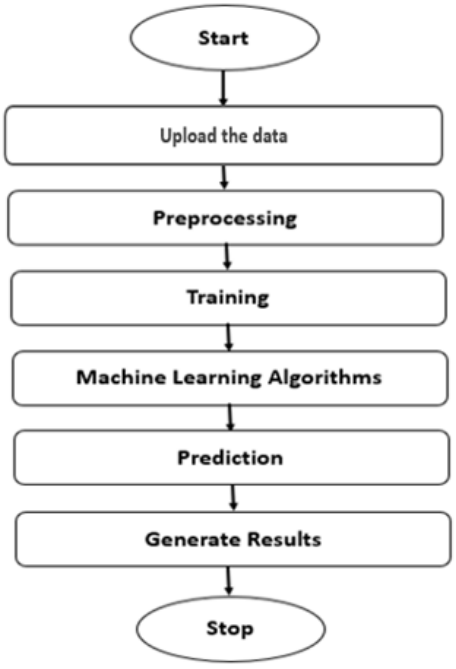
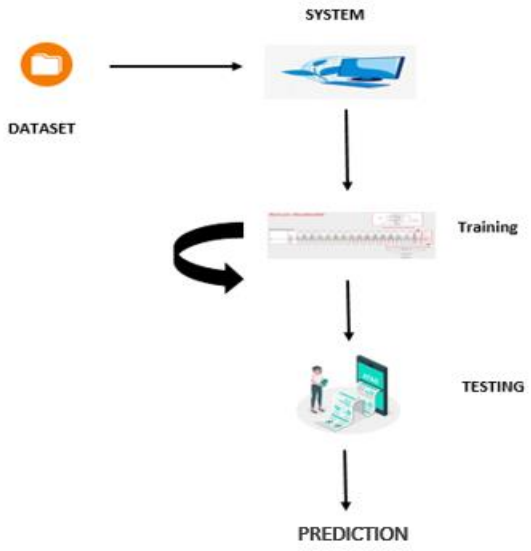


Figure 1: Block diagram

IV. Architecture



V. Existing Method

Due to a lack of knowledge about data visualisation, implementing machine learning algorithms in the existing system is a bit difficult. The present method depends on mathematical calculations to create models, which may be very time-consuming and difficult. We employ machine learning algorithms from the Scikit-Learn toolkit to circumvent all of this.

Disadvantages

- High complexity.
- Time consuming.

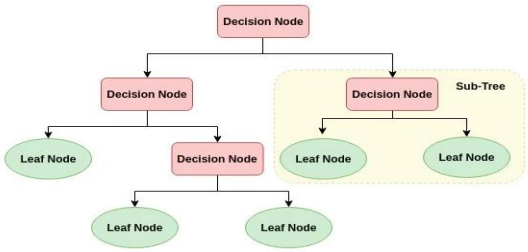
VI. METHODOLGY

The project has implemented by using below listed algorithm.

"Decision Tree"

An internal node represents a feature (or attribute), a branch represents a decision rule, and each leaf node shows the result in a decision tree, which resembles a flowchart. The root node is located at the top of a decision tree. Data subsets may now be created depending on attribute values. A tree is divided into discrete segments using recursive partitioning. This structure, which looks like a flowchart, helps in

decision-making. Just like a flowchart, it captures how individuals think. Decision trees are very straightforward to comprehend and apply.



The basic principle of any decision tree algorithm is as follows:

1. Using Attribute Selection Measures, select the best attribute to split the data (ASM).

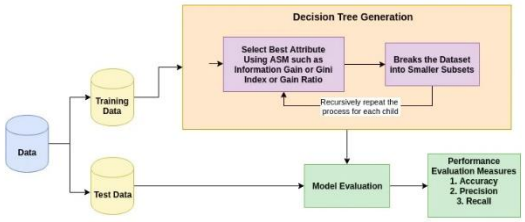
2. Divide the dataset into more manageable portions and designate that property as a decision node.

3. Repeats this procedure iteratively for each child to start the tree's growth until one of the conditions is satisfied:

All the tuples belong to the same attribute value.

There are no more remaining attributes.

There are no more instances.



"Extra Tree Classifier"

The Extra Trees Classifier ensemble learning approach relies heavily on decision trees. In order to avoid over-learning from the data and over-fitting, Extra Trees Classifier randomises some conclusions and subsets of data, similar to Random Forest. Let's examine a couple ensemble techniques as we go from high variance to low variance before finishing with Extra Trees Classifier.

Decision Tree (High Variance)

- When only one decision-making route is employed, a single decision tree frequently overfits the data it is learning from. When utilised with brand-new data, predictions from a single decision tree are often incorrect.
- Rough Forest (Medium Variance) Randomness added to random forest models prevents overfitting:
- building multiple trees (estimators)
- drawing observations with replacement (i.e., a bootstrapped sample)
- splitting nodes on the best split among a **random subset** of the features selected at every node

Extra Trees (Low Variance)

- In two important respects, Extra Trees differs from Random Forest: it does not bootstrap data (i.e., it samples without replacement), and nodes are divided on random splits rather than the ideal splits. With the aid of random characteristic picks, Random Forest creates several trees and divides nodes. The following is a summary of Extra Trees:
- It constructs numerous trees with the default setting of bootstrap = False, which samples without replacement
- A random subset of the features chosen at each node is used to separate the nodes.In Extra Trees, randomness doesn't come from bootstrapping of data, but rather comes from the random splits of all observations.

Extra Trees is named for (Extremely Randomized Trees).

" Random Forest Classifier"

Classification and regression issues are resolved using a machine learning approach known as a random forest. It employs ensemble learning, a method for

combining a number of classifiers in order to tackle challenging issues.

A random forest method has a wide variety of potential decision trees. With the use of bagging or bootstrap aggregation, the random forest method creates a "forest" that is then trained. The accuracy of machine learning systems is enhanced by bagging, an ensemble meta-algorithm. Based on the predictions produced by the decision trees, the (random forest) algorithm chooses the outcome. It makes predictions by averaging the results from different trees. The accuracy of the result increases with increased tree density.

The shortcomings of the decision tree approach are solved by the random forest method. Precision improves as dataset overfitting declines. Without a lot of parameters, it produces forecasts in packages (like Scikit-learn).

The Random Forest Algorithm has the following advantages over the Decision Tree Algorithm: • It provides a practical approach for addressing missing data; • It is more accurate; • It can produce a fair forecast without hyper-parameter change.

• At the node's splitting point in each random forest tree, a subset of characteristics is randomly selected to solve the problem of decision trees' overfitting.

In a random forest method, decision trees serve as the basic building blocks. Using a structure like to a tree, a decision tree is a decision assistance tool. Decision trees and the operation of random forest approaches will be covered.

The three parts of a decision tree are a root node, leaf nodes, and decision nodes. A decision tree approach separates the branches of a training dataset into smaller branches. Up until a leaf node is reached, this process is repeated. You can't divide the leaf node any further.

The decision tree's nodes represent the characteristics that are utilised to forecast the result. The leaves are connected by decision nodes. decision trees in their three various forms nodes.

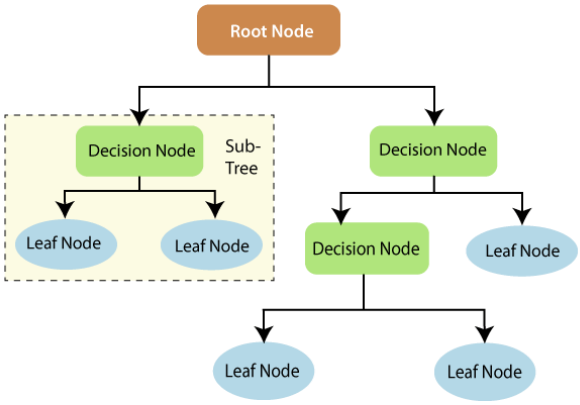


Fig:1

Decision trees' operation can be better understood in light of information theory. A decision tree is built on entropy and information gathering. We can better comprehend how decision trees are created by going through these key ideas.

Uncertainty may be measured using entropy. The amount of uncertainty in the target variable that is minimised given a collection of independent variables is measured as information gain.

Using independent variables (features) to learn more about a target variable is the idea of information gain (class). The information gain is calculated using the conditional entropy of Y and the entropy of the target variable (Y) (given X). The conditional entropy in this situation reduces the entropy of Y.

Information collection is used to train decision trees. It contributes to lessening the unease of these plants. An enormous quantity of ambiguity is eliminated when there is a considerable information gain (information entropy). Entropy and information gain are essential for branch splitting, a critical phase in the construction of decision trees.

See an example of a decision tree that is simple to understand. Let's imagine we want to predict a customer's propensity to purchase a mobile phone. He makes his choice in light of the phone's capabilities. This study may be presented using a decision tree diagram.

The decision's root node and decision nodes stand in for the aforementioned phone characteristics.Whether or not a purchase is made,

the leaf node indicates the result. The main selection factors are the cost, internal storage, and random access memory (RAM). The decision tree will appear as follows.

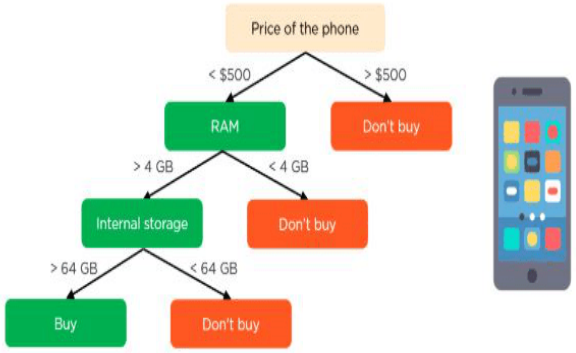


Fig:2

decision-tree application in random forest

The decision tree algorithm and the random forest technique are fundamentally different in that the latter randomly chooses the root nodes and clusters the nodes. The random forest employs the bagging technique to provide the required prediction.

Using numerous samples of data (training data) as opposed to only one is known as bagging. A training dataset's characteristics and observations are used to make predictions. The decision trees provide a range of outcomes depending on the training data that the random forest algorithm is given. The final output will be determined by which of these outputs has the highest rating.

Nonetheless, our original example may be used to show how random forests work. There will be several decision trees in the random forest as opposed to simply one. Count on their being just four decision trees in all. The training data, which includes of the phone's observations and characteristics, will be used to generate four root nodes in this case.

The root nodes indicate the four attributes that may influence a customer's decision (price, internal storage, camera, and RAM). The random forest will separate the nodes by randomly selecting the characteristics. The ultimate forecast will be chosen based on the outcomes of the four trees.

Most decision trees will choose the outcome in the end. If three trees indicate purchasing and one indicates not buying, the ultimate forecast will be buying. In this case, it is expected that the customer would buy the phone.

SVMs (Support Vector Machines) (Support Vector Machines)

The aim of the support vector machine is to find a hyper plane in an N-dimensional space (where N is the number of characteristics) that unambiguously classifies the data pointsalgorithm.

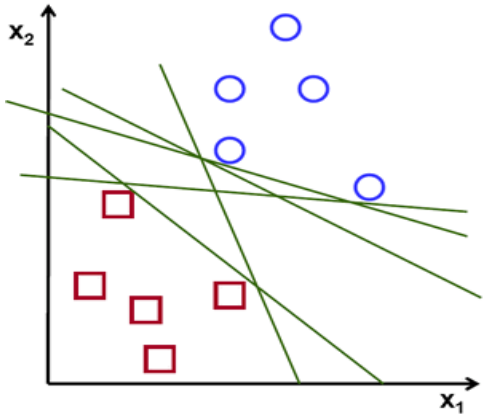


Fig:1

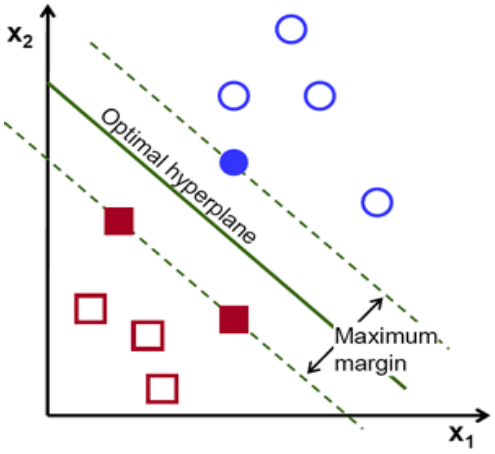


Fig:2

Possible hyper planes :

The two types of data points can be separated using a number of Hyper planes. Our objective is to find a plane with the largest margin, or the largest distance between data points from the two classes. The confidence with which future data points may be classified rises when the margin distance is increased since it provides some additional support.

Hyper planes and Support Vectors

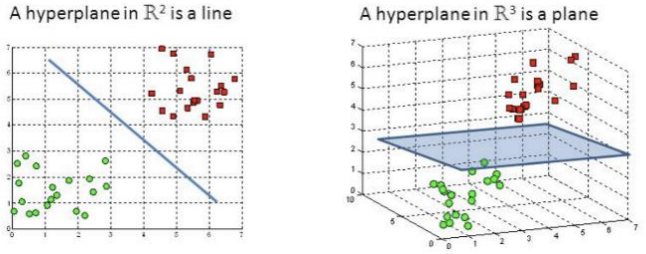


Fig:3

Hyper planes in 2D and 3D feature space

The categorization of the data components is aided by decision boundaries known as hyper planes. It is

$$\frac{\delta}{\delta w_k}\lambda \parallel w \parallel^2 = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k}\left(1 - y_i\langle x_i, w\rangle\right)_+ = \begin{cases} 0, & \text{if } y_i\langle x_i, w\rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

possible to classify data points that are on either side of thehyperplane. Also,

According on how many characteristics there are, the hyperplane will vary in size. A line is all that the hyper plane is if there are just two input features. Three input characteristics cause the hyper plane to collapse into a two-dimensional plane. When there are more than three elements, it gets increasingly difficult to imagine.

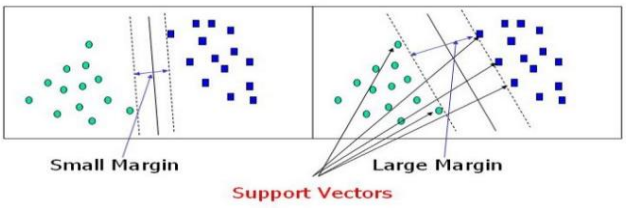


Fig:4

Support Vectors

Closer to the hyper plane data points known as support vectors have an impact on the hyper plane's location and orientation. These support vectors are used by us to increase the margin of the classifier. The location of the hyper plane will shift if the support vectors are eliminated. These are the guiding concepts that helped us create our SVM.

Large Margin Intuition

In logistic regression, the output of the linear function is compressed inside the range [0,1] by using the sigmoid function. We label the data as 1 or 0, depending on whether the compressed value exceeds a threshold value (0.5). The output of the linear function is taken into consideration by SVM. If the output value is larger than 1, it is attached to one class, and if it is less than 1, to another class. We acquire this margin, which has a broad range of values, by changing the threshold values in SVM to 1 and -1. ([-1, 1]).

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases}$$

the gradient and cost functions have been updated.
It is necessary to increase the space between the data. andthe SVM algorithm's hyperplane. To maximize the margin, a loss function called hinge loss is used.

What the hinge does (A function on the right can be used to represent a function on the left.)

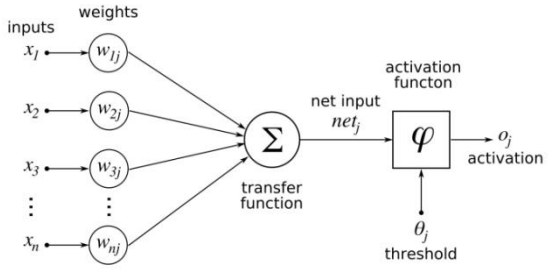
$$c(x, y, f(x)) = (1 - y * f(x))_+$$

There is no difference between the estimated and real values if all fall under the same banner: free. In that case, we next calculate the loss amount. We also supply the cost formula with a regularisation factor.
SVM's loss function
We may utilise partial derivatives with respect to the weights to calculate the gradients because we are aware of the loss function. We may modify our weights using the gradients.
Gradients
We only need to update the regularisation gradient when there is no misclassification, or when our model predicts the class of our data point accurately parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

No classification errors since the gradient update.
When there is a misclassification, or when our model forecasts the class of a data point wrongly, we add the loss together with the regularisation parameter to carry out gradient update.

In order to simulate how the human brain processes and analyses data, artificial neural networks (ANNs) were developed as a component of computing systems. It serves as the foundation of artificial intelligence (AI) and resolves problems that would be difficult or impossible by human or statistical standards. Because ANNs are self-learning, they can provide better outcomes as more data becomes available.
The hundreds or thousands of processing units, or artificial neurons, that make up an ANN are connected via nodesThese processing units consist of input and output units. In order to generate a single output report from the input units, which receive information in a variety of forms and structures based on an internal weighting system, the neural network attempts to learn about the information given. Backpropagation, also known as backward propagation of error, is a set of learning principles that artificial neural networks (ANNs) employ to improve the outcomes of their output, much as how humans utilise rules and guidelines to achieve results.An ANN must first complete a training phase before it can recognise patterns in any type of input, including text, audio, and visual patterns. The network compares its actual output to what it was anticipated to generate during this supervised phase, or the expected output. Using the backpropagation method, the discrepancy between the two results is corrected. In other words, the network modifies the weight of its connections between the units backward, from the output unit to the input unit, until the difference between the actual and projected outcome creates the least amount of error.



We always employ our Deep Neural Networks when we increase the number of layers in our ANN.
A deep neural network (DNN) comprises extra layers between the input and output levels of an artificial neural network (ANN). Although neural networks can take many different shapes, its fundamental components—neuron, synapses, weights, biases, and functions—remain constant.

VII. Results And Discussion

The project's flow and working method are shown in the following screenshots.

"HOME PAGE"

Here user view the home page of cyber-attack detection web application.

"ABOUT"

Here we can read about our project.

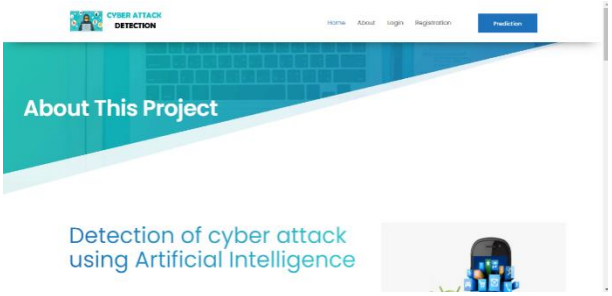


Fig:1

"REGISTER"

In the page, users need to register by entering his credentials.

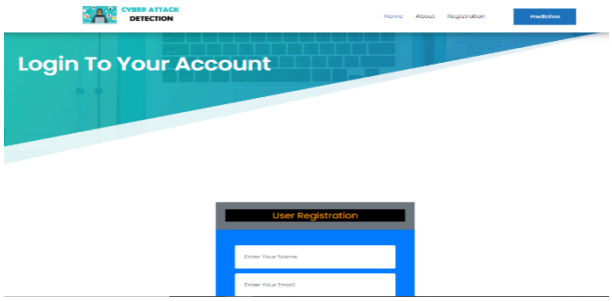


Fig:2

"LOG IN"

In the page, users has to enter the credentials to enter into the cyber-attack prediction.

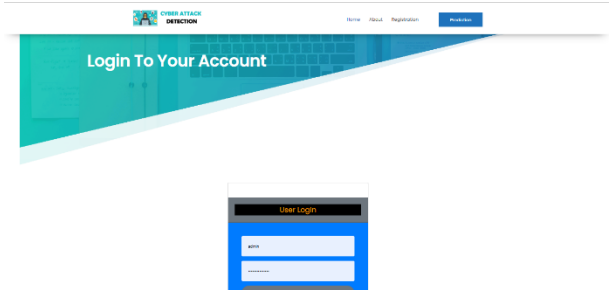


Fig:3

"LOAD"

In the load page, users can load the cyber dataset.

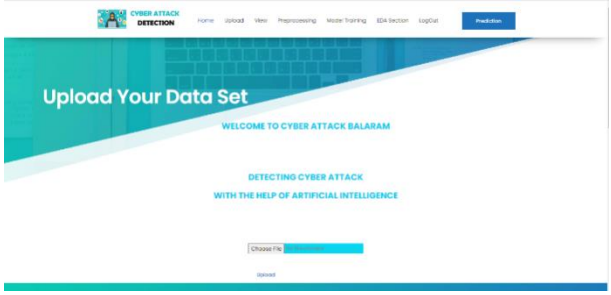


Fig:4

"VIEW"

Here we can see the uploaded data set.

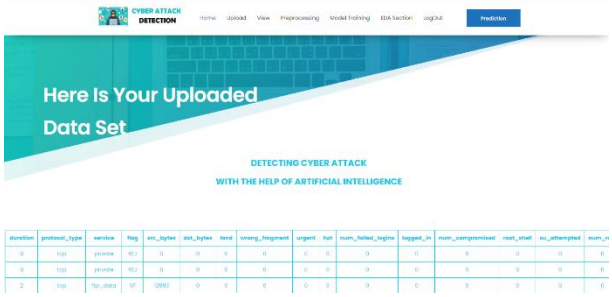


Fig:5

"PRE-PROCESS"

Here we can pre-process and split our data into train and test.

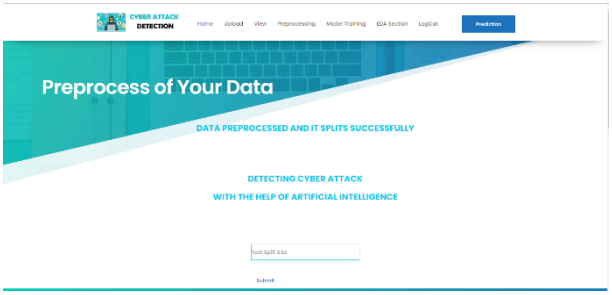


Fig:6

"MODEL"

Here we train our data with different ML algorithms.

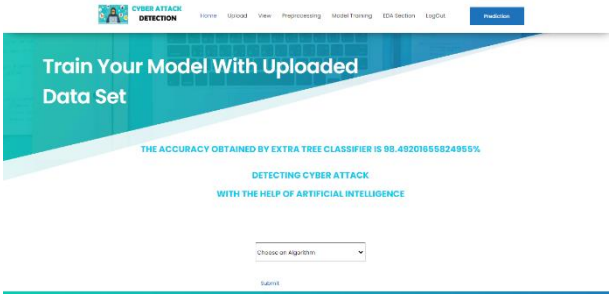


Fig:7

"PREDICTION"

This page show the detection result of the cyber-attack detection data.

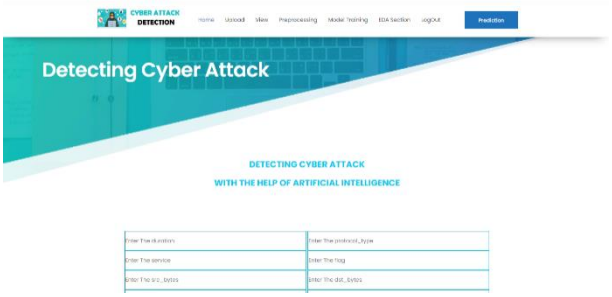


Fig:8

VIII. CONCLUSION

In this study, difficult discrete cyber-physical networks with a variety of local attacks disabled were used to evaluate the robust control consensus approach. It was shown that by isolating the attacked node and preserving system stability, the system may continue to function efficiently despite cyberattacks. It was proven that the system performance is enhanced by using a deep neural network with seven hidden layers, as was done in this study. Also, when a

deep layer network has a linear function, a recurrent neural network working in tandem with it performs better. As a result, the system might be thought of as being less complicated.Systems can look for patterns using the deep learning technique in order to stop similar assaults and react to changing behaviour. In conclusion, machine learning may considerably increase the efficiency, simplicity, and cost-effectiveness of cyber security. In order to prevent an attack from negatively affecting the behaviour of other agents, an attack is detected and separated by the control system based on the neural network's observations of the system's status. Further agent-based assaults, data mining, and machine learning techniques like support vector machine (SVM) algorithms or other types of neural networks like recurrent neural networks may be considered in future studies evaluating system performance improvements.

IX. REFERENCES

[1]. Kwon, Cheolhyeon, Weiyi Liu, and Inseok Hwang. ”Security analysis for cyber-physical systems against stealthy deception attacks.” In 2013 American control conference, IEEE (2013): 3344-3349.

[2]. Pajic, Miroslav, James Weimer, Nicola Bezzo, Oleg Sokolsky, George J. Pappas, and Insup Lee. ”Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators.” IEEE Control Systems Magazine 37, no. 2 (2017): 66-81.

[3]. Sheng, Long, Ya-Jun Pan, and Xiang Gong. ”Consensus formation control for a class of networked multiple mobile robot systems.” Journal of Control Science and Engineering 2012 (2012).

[4]. Zeng, Wente, and Mo-Yuen Chow. ”Resilient distributed control in the presence of misbehaving agents in networked control systems.” IEEE transactions on cybernetics 44, no. 11 (2014): 2038-2049.

[5]. Sun, Hongtao, Chen Peng, Taicheng Yang, Hao Zhang, and Wangli He. ”Resilient control of networked control systems with stochastic denial of service attacks.” Neurocomputing 270 (2017): 170-177.

[6]. Zhang, Haotian, and Shreyas Sundaram. ”Robustness of information diffusion algorithms to locally bounded adversaries.” In 2012 American Control Conference (ACC), IEEE (2012): 5855-5861.

[7]. Fu, Weiming, Jiahu Qin, Yang Shi, Wei Xing Zheng, and Yu Kang. ”Resilient Consensus of Discrete-Time Complex Cyber-Physical Networks under Deception Attacks.” IEEE Transactions on Industrial Informatics (2019).

[8]. Ozay, Mete, InakiEsnaola, Fatos Tunay YarmanVural, Sanjeev R. Kulkarni, and H. Vincent Poor. ”Machine learning methods for attack detection in the smart grid.” IEEE transactions on neural networks and learning systems 27, no. 8 (2015): 1773-1786.

[9]. Tianfield, Huaglory. ”Data mining based cyber-attack detection.” System simulation technology 13, no. 2 (2017): 90-104.

[10]. Pasqualetti, Fabio, Florian Dorfler, and Francesco Bullo. ”Attack detection and ¨ identification in cyber-physical systems.” IEEE Transactions on Automatic Control 58, no. 11 (2013): 2715-2729.

Cite this article as :

Y Rajesh, Ch. SivaKrishna, B. Jaswanth, J. SriHarsha, "Detection of Web based-Attacks Using Artificial Intelligence", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 2, pp.403-417, March-April-2023.