Analysis of Cybersecurity Attacks on Autonomous Vehicles using Machine Learning

Harshish Vataliya
MS Industrial Engineering
Northeastern University
Boston, MA
vataliya.h@northeastern.edu

***Abstract*—With daily advancements in autonomous technology, the field of artificial intelligence and deep learning applications in this automotive industry has noticed continuously increasing participation. The advancing applications however come with their own vulnerabilities. This report enlists the possible attacks on autonomous vehicles, but will focus on security attacks on the networks required for the vehicle's interaction with various entities. We use the KDDCup 99 dataset to address the problem by employing supervised and unsupervised Machine Learning Algorithms for detection and classification of the numerous attacks. While Supervised learning algorithm fail to detect the unknown attacks, the anomalous behaviour of these attacks can be detected by the unsupervised learning algorithms.**

I. Introduction

The past decade has seen a rapid growth in the Connected and Autonomous Vehicle (CAV) industry. This can be seen as a direct result of a continuously increasing market of consumers who have shown there interest and belief in the ”self-driving” vehicles. But, the real driver for this industry is the advancement in the computational technology and the artificial intelligence and machine learning capabilities. Despite the increasing user interests and investments in this industry, there is relatively less focus on the security and privacy of CAV data.

There are various levels of autonomous vehicles depending on the degree of autonomy. Vehicles with a low degree of autonomy give the driver more control and functionality for managing the vehicle. To achieve certain levels of autonomy, information about the environment is gathered entirely from on-board sensors without any active communication with other vehicles or the infrastructure. Furthermore, automated vehicles can communicate with each other and share information about the environment. Communication is not limited to communication between cars (vehicle-to-vehicle (V2V)), nor to communication between cars and the infrastructure (vehicle-to-infrastructure (V2I)). A cyber attack can start with control technology tools that are embedded in AVs such as electrical window controls, which are now controlled by engine control units (ECUs) as embedded systems. The ECU is one of the most important parts of a vehicle. An attacker can modify the programming code during design and implementation processing. Attackers target code in order to corrupt or degrade hardware performance, or to destroy information. A cyber attacker can configure the settings, modify code, and implant viruses and malware.

As mentioned in [5] by Jamal Raiyn, an autonomous vehicle is subject to attacks at one of these five technological tools that are embedded in them with respect to the level of autonomy: Sensor Networks, Cameras, Global Navigation Satellite Systems (GPS) and Radars, LiDARs and wireless communication. Securing the hardware tools from attacks is not the focus of this report and we shall focus on the attacks on the networks and communication paths.

II. Related Work

The past couple of years have brought numerous researchers into the field of artificial intelligence in autonomous vehicles. However, majority of the work has been limited to research and simulations, instead of real world implementation. The result of which is a lack of standardized approach in addressing the vulnerabilities of deep learning applications and cybersecurity in the autonomous systems.

The motive and the dataset preparation process was inspired from [1] where the researchers have explained the cybersecurity aspect of this field on the global level. They proposed a unified Modelling Language based framework and compared the attack detection performance with Naive Bayes Classifier and J48 Decision tree Algorithm with up to 97 per cent accuracy on the testing dataset after applying 10 fold cross validation on the training set with about 99 per cent accuracy.

In [2], the researchers presented a Machine learning based intrusion detection module highly specific to their approach on android malware detection. The dataset used is the CIC AWGM dataset provided by the Canadian Institute of Cybersecurity. The proposed algorithm is analyzed against ensemble learning algorithms such as decision trees, random forests, k-nearest neighbours, gradient boosting classifiers and bagging classifiers. All of the mentioned algorithms were tested for both multi-class classification and binary classification, where in the performances in the former case was considerably lower than for binary classification.

III. Data

Although there is a tremendous amount of on-going research in this field, there is a lack of specific cybersecurity dataset for CAVs. Therefore a more generic dataset, the KDDCup 1999 dataset is used, which is famous in the field of cybersecurity for Intrusion Detection and Network Analysis research and applications. It originated from the DARPA '98 IDS evaluation program and includes both labeled training data collected over seven weeks and unlabeled testing data collected over two weeks. Out of the 39 different attacks included in the dataset, the training set includes 22 attacks and the rest are a part of the testing dataset. For researchers with fewer resources, the dataset also includes 10 percent datasets for both categories.

The attack labels in the dataset belong to five categories namely Normal, Denial of Service (DoS), Probes, Remote-to-local (R2L), and User-to-Root (U2R). The DoS is an attack in which the attacker tries to make the target machine top providing service or resource access to system, while Probe represents surveillance and probing, and R2L refers to the unauthorized access while there is an illegal access from the remote machine to local one and represents that there is an unauthorized access to local superuser privileges by local unprivileged user.

Although the KDDCup is a clean dataset, that is, with no missing or faulty data points, the dataset is not representative of the real world scenario with high attack records and the normal records also do not represent the modern traffic. Hence, we use the NSL-KDD dataset which is an altered version of the KDDCup data set with over 125,000 training samples and more than 22,000 test samples. Similar to the KDDCup dataset, the NSL-KDD also has 20 percent subsets for contributors with limited resources. We further reduce this dataset by retaining only the relevant attacks for CAVs to keep the analysis focused on the motive. This reduction is based on the work of [1] where the the 39 attack types are categorized on the basis of possibility. This distribution is also provided in Table1. Therefore, the dataset is reduced only to the 14 highly possible attack types and the normal traffic.

Approach

The dataset includes 7 categorical features and 34 continuous features. We need to transform the dataset into numerical values with the help label encoders. We then split the datasets into two components X and y, where X contains all the features and y contains the respective labels. The next step is to bring all the features within the same scale before feeding them into the algorithms, which can be achieved through standard scaling the entire dataset. The KDDCup is a high dimensional dataset with 41 features and labels. For better performance results of the algorithms, we perform dimensionality reduction using Principal Component Analysis reducing the 41 features to two.

The target or the dependent variable here is the label of the attack which is categorical. Given the dataset is provided in two parts, as training and testing datasets, our first approach is implementing Supervised learning algorithms. The algorithms applied include k Nearest Neighbors, Decision Tree Classifier, and Support Vector Classifiers. The following section includes a brief description of these algorithms.

The k Nearest Neighbor Algorithm, as the name suggests, groups the data points together when they fall within a certain proximity. Although, kNN is very simple to understand and implement algorithm, it does not scale well with large datasets. The Decision Tree algorithm is an iterative process with binary splits based on rules identified from the data by the tree. The steps in a decision tree are simple : i) Select a test for root node. ii) Create branch for each possible outcome of the test. iii) Split instances into subsets. iv) One for each branch extending from the node. v) Repeat recursively for each branch, using only instances that reach the branch. vi) Stop recursion for a branch if all its instances have the same class. While decision trees are computationally inexpensive and easier to build with high classification performance, they usually tend to over fit and can introduce bias depending on the number or records for each classes in the dataset. However, Support Vector Machines have lesser tendency to over fit as they find the optimal hyper-plane in the dimensional space to separate the classes. Support vector Classifiers too are computationally inexpensive and widely used for high dimensional data but contrarily cannot be used for large datasets and do not work well for overlapping classes, as in our case.

2. Unsupervised Learning:

Since the testing dataset includes attacks previously unseen by the supervised learning algorithms, the resultant performance of these algorithms is very poor, as expected. This takes us to our second approach of Unsupervised learning to detect and classify the attacks. in Unsupervised learning, since we do not have the access to labeled data, the best way to understand the data is by forming clusters based on the features of the data which is also known as clustering. Two of the most popular clustering algorithms are KMeans and Density Based Spatial Clustering of Applications with Noise, better known as DBSCAN algorithm.

The KMeans Clustering algorithm uses distance metrics to form the clusters. It initiates by randomly placing centroids in the feature space and iteratively converges to the find the nearest cluster for each instance. The performance of KMeans can be analyzed either by Elbow method which measure the total within-cluster sum of squares or through the Silhouette score which compares the mean of intra-cluster distances with that of the nearest cluster for each instance.

The DBSCAN algorithm, similar to the KMeans algorithm, initiates clusters based on the distance metric and minimum number of samples. The algorithm requires a parameter called epsilon which is basically the radius for an instance at the center of a cluster. The algorithm constitutes of three categories for the instances, core points, border points and outliers or noise. The instances with minimum samples within its proximity of epsilon are known as core points, where as the instances with none of the samples within epsilon distance are labelled as noise. The instances which do have samples in their proximity but do not reach the minimum samples required to initiate a cluster are labelled as the border points. While this algorithm is capable of handling data with high dimensionality and is scalable to large datasets, it foes not work well with data having clusters with varying densities

IV. Experiments and Results

The supervised learning algorithms are trained with 5-fold cross validation on the training sets and analyzed against the testing sets. The outcome, as expected, was very poor performance from all the algorithms as supervised learning algorithms need to learn about the data before its application. The incapability of these algorithms to identify unknown data is not suitable for the field of cybersecurity where the attacks evolve with evolving firewalls. However, when the algorithms were exposed to all the possible attacks, each and every algorithm achieved accuracy higher than 90 percent.

The KMeans and DBSCAN algorithm performed fairly well for the training and testing datasets, but their performances are highly dependent on their hyper-parameters, such as the values of K in KMeans and the values of minimum samples and epsilon with the dimensionality for the DBSCAN algorithm.

The experimental implementation of this report can be found at: https://github.com/vataliya/cav$_{s}ecurity_{a}ttacks$

V. Summary and Future Work

Despite the development of different configurations of connected vehicles, they are still vulnerable to various security issues and there are various automotive attack surfaces that can be exploited [4]. The various approaches presented here have numerous shortcomings which can be overcome with the implementation of deep unsupervised learning algorithms such as deep neural networks or Long-Short Term Memory (Recurrent Neural Networks) with Auto-encoders. Another interesting approach is applying adversarial reinforcement learning for detecting, classifying and taking counter measures before the attack causes considerable damage to the vehicles, the passengers or the surroundings in any way.

VI. References

[1] Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles Qiyi He , Xiaolin Meng, Rong Qu and Ruijie Xi Received: 2 June 2020; Accepted: 5 August 2020; Published: 7 August 2020

[2] Malware Detection in Self-Driving Vehicles Using Machine Learning Algorithms Seunghyun Park and Jin-Young Choi, Guest Editor: Hsing-Chung Chen Received 26 July 2019; Revised 19 October 2019; Accepted 19 November 2019; Published 17 January 2020

[3] Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey Yirui Wu , Dabao Wei, and Jun Feng, Academic Editor: Xiaolong Xu Received 7 May 2020; Revised 26 June 2020; Accepted 20 July 2020; Published 28 August 2020

[4] ”Comprehensive experimental analyses of automotive attack surfaces” by Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, et al.

[5] ”DATA AND CYBER SECURITY IN AUTONOMOUS VEHICLE NETWORKS” by Jamal Raiyn Computer Science Department, Al Qasemi Academic College ; Baqa El Gharbia, Israel ; raiyn@qsm.ac.il

[6] KDD Cup 1999 Data
http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[7] NSL-KDD Dataset
https://www.unb.ca/cic/datasets/nsl.html

[8] CIC AWGM dataset
https://www.unb.ca/cic/datasets/ddos-2019.html

[9] ”When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets” by Hang Yin and Christian Berger Department of Computer Science and Engineering, University of Gothenburg, Gothenburg, Sweden

[10] DDoS attack detection: A key enabler for sustainable communicationin internet of vehicles by Hafiz Husnain Raza Sherazia, Razi Iqbalb, Farooq Ahmadc, Zuhaib Ashfaq Khand,Muhammad Hasanain Chaudaryca Department of Electrical and Information Engineering, Politecnico di Bari, Bari 70125, Italy

[11] Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network by Yanqing Yang , Kangfeng Zheng , Chunhua Wu and Yixian Yang Received: 18 March 2019; Accepted: 30 May 2019; Published: 2 June 2019

[12] Survey and Classification of Automotive Security Attacks by Florian Sommer , Jürgen Dürrwang and Reiner Kriesten Institute of Energy Efficient Mobility (IEEM), University of Applied Sciences, Moltkestrasse 30, 76133 Karlsruhe, Germany Received: 1 April 2019; Accepted: 16 April 2019; Published: 19 April 2019

[13] Securing Connected Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and The Way Forward by Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha

[14] Security Analytics: Adapting Data Science for Security Challenges by Rakesh Verma, University of Houston ,Houston, Texas

[15] Simulation and Analysis of DDoS Attack on Connected Autonomous Vehicular Network using OMNET++ by Tauheed Khan Mohd, Subhrajit Majumdar, Akshay Mathur, and Ahmad Y. Javaid EECS Department, The University of Toledo, Toledo, OH