Mark Demore, 2d Lt

CSCE623 – Dr. Borghetti

Project Final Draft

**ABSTRACT** – GPS Spoofing has become more common, as the ease to implement such an attack increases. All USAF assets rely on GPS data for navigation and timing, and are vulnerable to spoofing attacks. A data focused approach would provide a flexible, lightweight solution that could be easily implemented on a variety of systems. Machine learning techniques provide a promising way to develop such a system. Logistic regression, naïve Bayes, and random forests classifiers were trained on simulated spoofing data to detect a spoofing attack. All of these methods showed promise when given appropriate training data, and random forests proved to be the best solution with high recall, even outperforming a deep learning method implemented in previous research. Future work is needed to test the approach with more subtle spoofing attacks.

1. Introduction

The proliferation of Software-Defined Radio (SDR) and other Commercial Off-the-Shelf (COTS) systems has led to increased vulnerability of United States Air Force (USAF) systems through the spoofing of Global Positioning System (GPS) signals. USAF assets (drones, airplanes, ground systems) rely on GPS signals to coordinate timing and for navigation. While military GPS signals are encrypted and more difficult to spoof, many systems also use civilian GPS messages. The MIL-STD-1553 avionics bus coordinates traffic between systems on USAF assets. It times messages between a bus controller (BC) and the different components (GPS, INS, Flight Computer), or remote terminals (RTs). The most effective lightweight solution to counter GPS spoofing would be in a bus monitor for MIL-STD-1553 that can parse the messages between the avionics and detect GPS spoofing, using a machine learning approach.

This research strives to answer the following questions: Can a supervised machine learning algorithm classify MIL-STD-1553 messages as spoofed based on observations from the bus? The central hypothesis is that a machine learning algorithm can correctly classify MIL-STD-1553 messages with statistical significance using bus traffic from other avionics components as features, 13 in total. It is also likely that a more drastic spoofing attack, one with more deviation in position from true position, will be easier to detect and have higher accuracy. The features collected in the data are: GPS Latitude, GPS Longitude, GPS Altitude, ADC Altitude, ADC Air Speed, EFIS Angle of Attack, INS X-Velocity, INS Y-Velocity, INS Z-Velocity, INS X-Acceleration, INS Y-Acceleration, and INS Z-Acceleration. The Inertial Navigation System (INS) data will likely have the most significance in classifying when compared to the GPS data .The research objective is to develop a classification model that can correctly classify spoofed GPS messages on the MIL-STD-1553 bus with recall of 90% or greater. Recall will be the main measure of success, since false negatives will result in navigation errors while a false positive will simply result in further verification and monitoring of system data.

The data used in this project has been procured from a previous student’s thesis work. It was generated using the Air Force Research Lab’s (AFRL) Avionics Vulnerability Assessment System (AVAS). The data was collected from 5 flights of lengths of about 4 hours, for a total of approximately 700,000 observations per flight, each collecting all of the aforementioned features. Spoofing occurred for periods of 3 minutes and was recorded as truth data for use in supervised learning methods, with equal portions of the flight being spoofed and not spoofed.

1. Related Work

A number of papers have been published on the application of machine learning to anomaly detection in time-based systems. Marvin attempted a deep learning approach to detect spoofing with mixed results, likely due to issues with the data and the additional complexities of a neural network application [5]. Genereux, et al. has used the timing of messages over the 1553 bus to detect anomalies and intrusion using a simple histogram approach [7]. Other work includes the use of Support Vector Machines to detect GPS spoofing using the signals themselves and a supervised approach [8]. More generalized research includes the use of Markov models and relative probabilities [1], and applications of deep learning to detect ADS-B spoofing [4] and network intrusion [2]. The aforementioned research is representative of the applicability of machine learning methods to this problem. This project differs mostly in the data that is being used, messages from the MIL-STD-1553 bus. The advantage of using 1553 data is that developing a simple bus monitor with the detection algorithm would provide a lightweight solution that can be implemented on any USAF asset.

1. Methodology

The data has been collected using the AVAS system in tandem with AFRL’s Vampire, an FPGA-based bus monitor, that sent the 1553 messages as UDP packets to a Windows 10 machine running a Python script to save the data to a CSV file. The flights were flown by hand in level and banked flight orientation, as well as with varying altitudes with climb and descent maneuvers. The spoofing attacks implemented drift and hold position attacks of varying displacement rates (5, 10, 50, and 100 m/s). The attack simulated drift in the X, Y, and Z axes. The data is then parsed using an XML file encoded from the 1553 bus Interface Control Document (ICD), which details how the data in the message is wrapped. This is necessary because each platform has a different ICD, and the intermediate step will allow the solution to be flexible and fit any system once an XML is made for its specific bus architecture. Messages that do not contain information about the 13 features of interest are dropped upon parsing and left out of the data.

Three different classification methods attempted to discern spoofing attacks: logistic regression, naïve Bayes, and random forests. More specifically, logistic regression included the use of 5-fold cross validation, the naïve Bayes classifier used a Gaussian model, and the random forests classifier consisted of 100 trees. A total of 10 different instances were trained and tested with each of the three models. All of these instances used an 80/20 split for train/test data. First, each model trained and tested on the attacks individually: fixed point, 5m/s drift, 10m/s drift, 50m/s drift, and 100m/s drift. Next, the modesl were trained on all but one attack and tested on the unseen attack, rotating through all of the attacks being unseen. The unseen attack test also broke down into an approximately 80/20 train/test split, since there are a total of 5 different attacks and each contains approximately the same number of observations. Finally, the models were trained on 80% of the total data and tested on the remaining 20%, combining all attacks in the training and testing data. Equal representation of each attack was assured by splitting the individual attacks’ data into train and test sets and combining them into the ‘combined’ train and test set respectively.

After the models trained, they were tested on the remaining data and collected results in the form of precision, recall, and F1 score. Recall is the primary measure of success in this application, since a false positive is preferable to a false negative, but precision and F1 score also provide meaningful metrics in assessing the models. The naïve Bayes model serves as a baseline for the other two when judging the effectiveness of a machine learning approach to the problem. Ultimately, the success of the models will be ranked, and compared to the deep learning method employed by Marvin [5].

1. Results

The results verify the practicality of a supervised machine learning approach to the problem, and show promise, particularly in the use of a random forests classifier. They also emphasize the importance of having quality training data, as unseen attacks provided the greatest obstacle for the models. As shown in Table 1, the random forests classifier performed perfectly in each scenario, while logistic regression and naïve Bayes performance deteriorated as the attacks became more subtle. Logistic regression and naïve Bayes also performed mediocrely when trained and tested on all attacks.

**Table 1: Individual Attack Scenarios**



LCV = Logistic Regression with Cross-Validation, GNB = Gaussian Naïve Bayes, RFC = Random Forests Classifier

More interestingly, all of the models underperform in the face of an unseen attack, highlighted in Table 2. Surprisingly, the models perform worse when the fixed point attack is withheld than they do when the most subtle attack is withheld. However, the trend of lower model performance with more subtle drift rates holds for the remaining attacks.

**Table 2: Unseen Testing Attack Scenarios**



LCV = Logistic Regression with Cross-Validation, GNB = Gaussian Naïve Bayes, RFC = Random Forests Classifier

When compared to the deep learning model on the same data set and scenarios, it is interesting to note that the random forests classifier outperforms the multivariate long-short term fully convolutional network, shown in Table 3. While the increase in model recall is marginal, it is worth noting that the computational requirements for the classical machine learning model are far lower than that of the deep learning model, and it may provide a more flexible solution. Moreover, the random forests classifier may outperform the deep learning model more drastically when faced with more advanced spoofing attacks.

**Table 3: Deep Learning Model Comparison**



RFC = Random Forests Classifier, MLSTM = Multivariate Long-Short Term Fully Convolutional Network

1. Conclusion & Future Work

The results obtained show the promise of a supervised machine learning approach, implemented in a MIL-STD-1553 bus monitor, in detecting a GPS spoofing attack. The random forests classifier had perfect results when trained using proper data. These results also emphasize the importance in obtaining accurate, realistically spoofed data in implementing such a solution. There are potentially severe limitations of the models when attempting to identify an unseen spoofing attack. The data also proves that a less drastic spoofing attack is harder for the models to detect, as there is less contrast in the readings to determine anomalous behavior. Moreover, a classical machine learning model seems to provide better intuition at less cost than a deep learning model. However, further testing is necessary to verify this outcome.

Future work must be conducted in order to verify these results. Specifically, much more advanced and subtle spoofing attacks must be implemented in the training and test data. The attacks implemented in the data used are much simpler than would be realistically employed. Also, the data should reflect a response by the aircraft’s flightpath as a result of spoofed navigation data. The data used in this research does not reflect any deviation in the aircraft’s flightpath with the presence of conflicting GPS data, but in a real attack the aircraft should begin following the spoofed route. With more realistic data, the same method can be applied and should reproduce these results.

1. References

[1] Du, Y., Wang, H., & Pang, Y. (2004). A Hidden Markov Models-based anomaly intrusion detection method. Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 5, 4348–4351. https://doi.org/10.1109/wcica.2004.1342334

[2] Van, N. T., Thinh, T. N., & Sach, L. T. (2017). An anomaly-based network intrusion detection system using Deep learning. Proceedings - 2017 International Conference on System Science and Engineering, ICSSE 2017, 210–214. https://doi.org/10.1109/ICSSE.2017.8030867

[3] James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An Introduction to Statistical Learning - with Applications in R | Gareth James | Springer. Retrieved from https://www.springer.com/gp/book/9781461471370%0Ahttp://www.springer.com/us/book/9781461471370

[4] Ying, X., Mazer, J., Bernieri, G., Conti, M., Bushnell, L., & Poovendran, R. (2019). Detecting ADS-B Spoofing Attacks Using Deep Neural Networks. 2019 IEEE Conference on Communications and Network Security, CNS 2019, 187–195. https://doi.org/10.1109/CNS.2019.8802732

[5] Marvin, J. M. (2019). Detecting GPS Spoofing with Deep Learning. https://doi.org/10.1109/TE.1962.4322266

[6] Wang, J., Tu, W., Hui, L. C. K., Yiu, S. M., & Wang, E. K. (2017). Detecting Time Synchronization Attacks in Cyber-Physical Systems with Machine Learning Techniques. Proceedings - International Conference on Distributed Computing Systems, 2246–2251. https://doi.org/10.1109/ICDCS.2017.25

[7] Genereux, S. J. J., Lai, A. K. H., Fowles, C. O., Roberge, V. R., Vigeant, G. P. M., & Paquet, J. R. (2020). MAIDENS: MIL-STD-1553 Anomaly-Based Intrusion Detection System Using Time-Based Histogram Comparison. IEEE Transactions on Aerospace and Electronic Systems, 56(1), 276–284. https://doi.org/10.1109/TAES.2019.2914519

[8] Semanjski, S., Muls, A., Semanjski, I., & De Wilde, W. (2019). Use and validation of supervised machine learning approach for detection of GNSS signal spoofing. 2019 International Conference on Localization and GNSS, ICL-GNSS 2019 - Proceedings, 1–6. https://doi.org/10.1109/ICL-GNSS.2019.8752775