Mark Demore, 2d Lt

CSCE823 – 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. Both classical and advanced machine learning models were trained on simulated spoofing data to detect a spoofing attack. Only a few classical classifiers showed promise when given appropriate training data, and random forests proved to be the best solution with high recall. However, advanced models, including Long-Short Term Memory and Convolutional Neural Networks gave problematic results. Future work is needed to test the dataset with deep learning models and pilot experiments in alternate approaches.

I. 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 a bus monitor for MIL-STD-1553 that can parse the messages between 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, 17 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, Velocity-North, Velocity-East, Velocity-Down, Attitude-Roll, Attitude-Pitch, Attitude-Yaw, Barometer-Altitude, IMU-Delta Theta (DT)-North, IMU-DT-East, IMU-DT-Down, IMU-Delta Velocity (DV)-North, IMU-DV-East, IMU-DV-Down. The Inertial Measurement Unit (IMU) 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 set of real flight data curated by the Autonomy and Navigation Technology (ANT) center. That flight data was used in tandem with the ANT Center's Fly program, which generates simulated sensor readings with error from the flight data. The data was collected from 4 flights, totaling over 8 hours of data and over 5 million observations. The spoof was implemented when combining the various sensor readings from Fly into a single Comma Separated Value (CSV) format, altering the GPS readings to simulate the attack. The spoof occurred in minute intervals, and the class (spoofed or unspoofed) was recorded alongside the rest of the data.

II. Related Work

A number of papers have been published on the application of machine learning to anomaly detection in time-based systems. All of these models address a simple binary classification problem, distinguishing between normal system behavior and anomalous system behavior [3]. 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]. Similarly, message timing data has been used to detect an attack in a variety of cyber-physical systems [6]. Other work includes the use of Support Vector Machines to detect GPS spoofing using information from the signal characteristics and a supervised approach [8]. More generalized research includes the

use of Markov models and relative probabilities to detect anomalies in time series data [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 and other anomalous behavior in time series data. 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 trained detection algorithm would provide a lightweight solution that can be implemented on any USAF asset.

III. Methodology

The flight data that was collected in the original dataset is very similar to the data that would be collected directly from the MIL-STD-1553 bus, but was instead collected using the ANT Center's Scorpion system. As such, it was used with the Fly program to generate sensor data to ensure reproducibility. All data was recorded using the WGS-84 coordinate system. The flight data and sensor readings were recorded in log files which were then parsed and combined in a CSV file. Since the flight data was recorded every millisecond, it was downsampled, based on nearest timestamp, to match the observation per second output from Fly. As a result, each observation in the final dataset represents one second of flight time. In between the combination of log files into the singular CSV, the spoofing attack was implemented. The attacks vary in their drift from the true position, the largest of interest is a drift rate of 5 meters per second in a single axis, although larger drift rates, up to 1000 meters per second, were tested as well. A future attack to be implemented will shift the position readings in time. Once the data was recorded, it was separated into its 18 features, 3 of which were susceptible to the attack; GPS-Latitude, GPS-Longitude, and GPS-Altitude. Finally, every feature in every observation was normalized to represent a value between 0 and 1 to assist the model and exacerbate smaller changes in feature values.

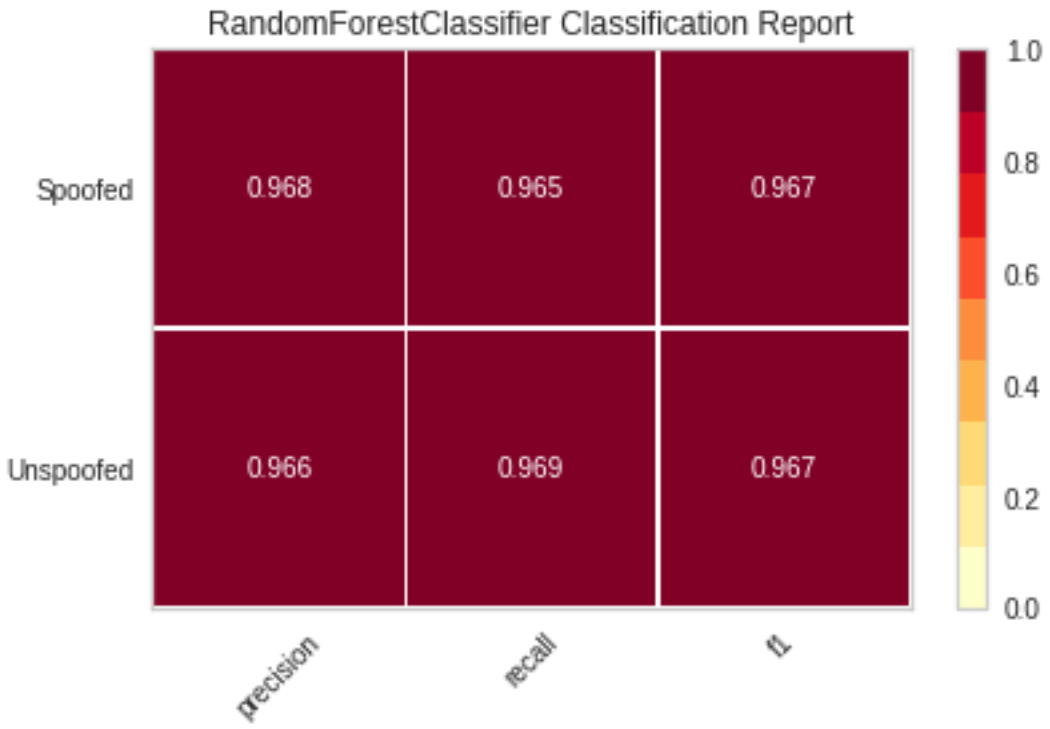
Three different classical classification methods attempted to discern spoofing attacks: random forests, k-nearest neighbors, and support vector machines. More specifically, the random forests classifier consisted of 100 trees, and k-nearest neighbors used 5 neighbors. Six advanced classification methods were also evaluated with the dataset: two different basic multilayer perceptrons (MLP), a convolutional neural network (CNN), a multivariate long-short term memory (MLSTM) model, an MLSTM with dropout, an MLSTM with convolution, and a previous students MLSTM Fully Convolutional Network (FCN). These models were evaluated with tensors ranging from 3 timesteps to 1000 timesteps, and similarly variable batch sizes, with the potential for larger batches with less timesteps and the necessity of smaller batches with more timesteps in the tensors due to memory constraints. While small drift rates are of greater interest, because they will be more difficult to detect, the performance of the models warranted testing with larger drift rates, of 100m/s and even 1000m/s. All of these instances used an 80/20 split for train/test data. The models were trained and tested on the same drift rate in latitude for a single flight, and also for all flights combined. There was equal representation of both spoofed and unspoofed observations across the data, as well as in the training and testing data.

After the models trained, they were tested on the remaining data and collected results. For the classical models, the results were compiled 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 classical models serve as a baseline to the advanced methods, as well as a proof of concept and validity. An effective model is measured by a recall rate of more than 90%. The advanced methods, given their poor performance with the existing dataset and current parameters, were only evaluated using loss and accuracy. Since the classification is binary, the loss function used for training and evaluation was binary crossentropy. Similar to recall, an accuracy of more than 90% would indicate an effective model.

IV. 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 or k-nearest neighbors classifier. They also emphasize that an altered dataset or better adapted models may be needed to make use of advanced machine learning techniques. As shown in Table 1, the random forests classifier performed very well with a drift rate of 5m/s, and was very comparable to the results obtained using the k-nearest neighbors classifier. The support vector classifier only output a single class, an issue that carried over into some of the advanced models.

Table 1: Random Forests Performance



1min attack duration, 5m/s drift in latitude

The contrast in performance between two of the classical methods and that of the more advanced models is the most interesting outcome. Figure 1 shows the loss and accuracy for the basic multilayer perceptron classifier over 100 training epochs. While the trend of these plots is indicative of a successful model, the accuracy only grows from a high 49% to a low 50%.

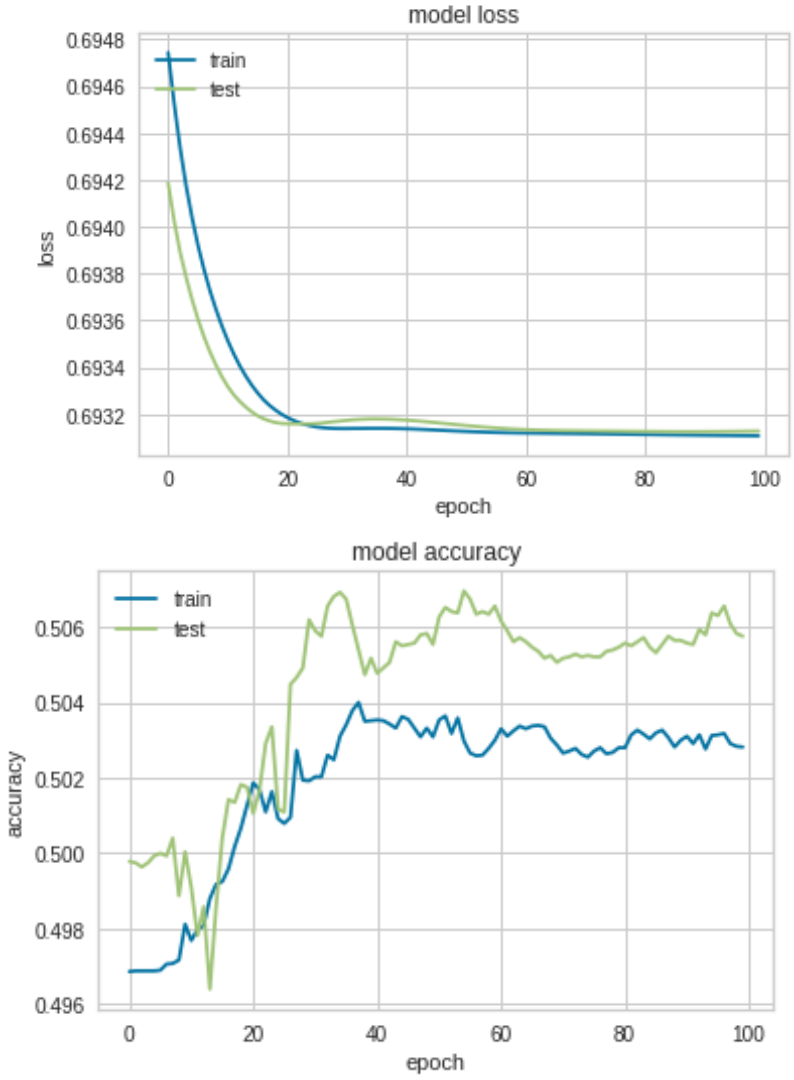


Figure 1: Performance of the simple Multilayer Perceptron Classifier

The accuracy for the remaining deep learning models share in this pitfall, although their loss and accuracy over time are more haphazard with each epoch, with sudden shifts and a divergence between the train and test sets. Figure 2 below provides an example, the loss and accuracy plots for a Long-Short Term Memory model. The results from these models are similar to the results from the Support Vector model, only outputting a single class. A variety of hyperparameters were adjusted, as well as different normalization methods with the data, and the results remained the same in spite of this, across the models. The spoof was implemented with up to a 1000m/s drift rate, and achieved similarly lackluster results.

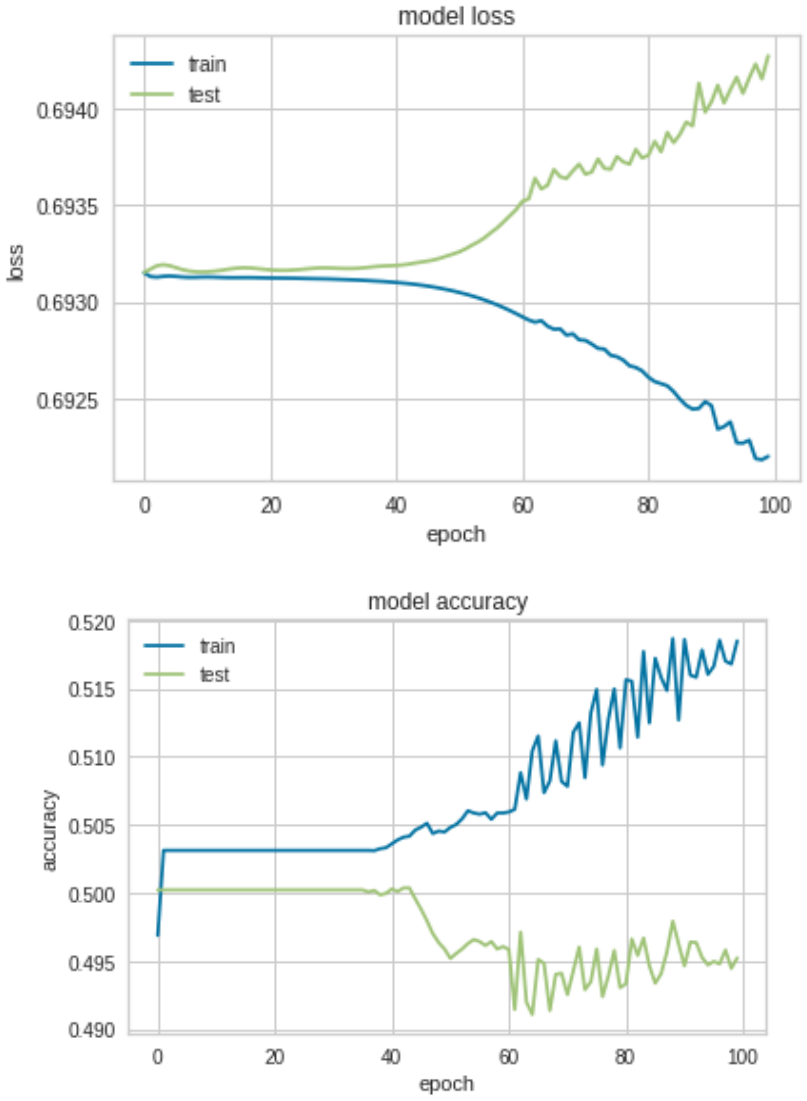


Figure 2: Performance of the MLSTM Model

The Multivariate Long-Short Term Memory Fully-Convolutional Network implementation yielded no change in loss nor accuracy over training, and will likely need further adaptation to work with the new dataset. These results show that these models may not be as well equipped to handle the nature of this task, and that the data may need to be modified further, such as transforming the coordinate system, in order to make small differences more discernable. However, the ability of classical methods to reliably detect spoofing show potential with the new dataset.

V. Conclusion & Future Work

The results obtained show the feasibility of a machine learning approach, implemented as a MIL-STD-1553 bus monitor, in detecting a GPS spoofing attack. The random forests classifier had exceptional results when trained using the new data. These results also emphasize the importance in obtaining accurate, realistically spoofed data in implementing such a solution and that smaller attacks within the range of error between observations are much more difficult to detect. There are potentially severe limitations of the models at the moment, but the models and data set can likely be adapted to yield more meaningful results. Moreover, as was noted in previous research, a classical machine learning model may provide better intuition at less cost than a deep learning model. However, further testing is necessary to verify this outcome. Using a measure of a recall rate of at least 90%, the random forests classifier and k-nearest neighbors classifier succeed in these initial tests. All other models failed in initial tests.

Future work must be conducted in order to verify these results. Specifically, the data should be modified in order to exacerbate the difference in attacks and the models should be verified with an alternate dataset. An area of interest in particular would be using a deep learning method for anomaly detection, such as autoencoders. By training an autoencoder model on normal flight data, it can learn the behavior and output a measure of error during training to use as a threshold when given new data to detect anomalies, or spoofs. With further analysis of the shortcomings of these existing models with the existing data, a determination can be made as to whether or not this pivot in approach will be necessary. Nonetheless, the performance of classical methods are evidence that further pilot experiments will be necessary and that the existing data is valid in at least some cases.

VI. 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

[9] Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, 237–245. https://doi.org/10.1016/j.neunet.2019.04.014

[10] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning.* 2016. http://www.deeplearningbook.org.