

MACHINE LEARNING METHODS FOR DIGITAL SIGNAL SEPARATION

by

Keith Michael Filler

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2025

©COPYRIGHT

by

Keith Michael Filler

2025

All Rights Reserved

DEDICATION

I dedicate this thesis to my family, for their endless support on my college journey.

ACKNOWLEDGEMENTS

The author acknowledges Dr. Bradley M. Whitaker, Dr. Ross K. Snider, Dr. Brock J. LaMeres, and Dr. Todd Kaiser. The author also acknowledges Maximus Marceau, Skyler Reid, Keith D. Mecham, Noah Netz, and Stephen Ude for their collaboration on this research. In addition, the author acknowledges the faculty and staff of the Departments of Electrical and Computer Engineering, Physics, and Science Education at Montana State University. Lastly, the author acknowledges his student colleagues within these departments.

This research is part of an ongoing collaboration between Idaho National Laboratory (INL) and Montana State University (MSU) to explore applications of machine learning to enhance protection of process control systems from cybersecurity threats. Funding for this collaboration is provided through the INL Laboratory Directed Research & Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

VITA

Keith Michael Filler was born in Littleton, CO, on 22 May 1990. Keith, and his two sisters, were raised in Kalispell, MT by David J. and Karie L. Filler. He attended Flathead High School and graduated with a diploma in June 2008. Keith attended Flathead Valley Community College (FVCC) from September 2009 to May 2012. Initially, Keith earned an Associate of Arts and an Associate of Science degree from FVCC. Later, he completed the academic year 2014-2015 at FVCC studying welding technology, earning a Certificate in Welding Technology. In September 2015, Keith enrolled at the University of Montana. Keith completed a Bachelor of Arts in Physics in May 2018.

TABLE OF CONTENTS

1. INTRODUCTION	1
General	1
Ethernet Communication	1
Machine Learning	2
Summary	3
2. BACKGROUND	4
Machine Learning Literature Review	4
Communication Technology Literature Review	6
Machine Learning Applications in Communication Technology Lit- erature Review	9
3. SUPERVISED MACHINE LEARNING FOR MODBUS COMMUNICA- TION PROTOCOL DECODING	12
Introduction	12
Methods	14
Data Collection	14
Manual Decoder	17
Machine Learning Decoder	17
Results	19
Manual Decoder	19
Machine Learning Decoder	19
Conclusion	22
4. LSTM FOR DIGITAL SIGNAL SEPARATION	23
Introduction	23
Methods	23
Data Collection	24
LSTM for Signal Separation	26
Results	27
Conclusion	31
5. THESIS CONCLUSION	33
REFERENCES CITED	38

LIST OF FIGURES

Figure	Page
1. Figure 1 Experiment layout for the initial voltage measurements from a MODBUS communication scheme. These measurements are then converted into usable data for training the machine learning models.	14
2. Figure 2 Illustration of the process of converting data frames to modbus log entries. The data creation process involved alternating transmissions (Tx) and responses (Rx) between two communication nodes. There is a brief pause (transmitting only zeros) between corresponding Tx/Rx signals, and a longer pause after the complete Tx/Rx pair. This transmit/receive procedure was repeated 88,485 times, at which point one billion samples were collected by the oscilloscope. The one billion samples include information about 176,970 traces plus the zero-only pauses between traces.....	15
3. Figure 3 Trace structure showing address packet, function code packet, and the first data packet within the trace. Note that each packet contains 11 bits and about 110 samples, meaning that the signal has approximately 10 samples per bit.	16
4. Figure 4 Data structure of a MODBUS packet.	17
5. Figure 5 An example of analyzing a determined binary sequence. The first step in this procedure is to account for the little endian style encoding scheme typical to MODBUS data. This requires the initially determined binary sequence to be flipped from LSB first to MSB first. This new binary sequence is then converted to the hexadecimal equivalent and compared to the MODBUS log data.	18
6. Figure 6 Table of hyperparameters for KNN training procedure.....	19
7. Figure 7 Comparison of the determined hexadecimal values to the recorded MODBUS data log values. The results show a 100 percent accuracy in the manual decoding algorithm.....	19
8. Figure 8 Confusion matrix. Note that messages with index greater than 214 were predicted by the algorithm to be a message not present in the test dataset. The presence of a strong diagonal indicates a successful classifier.	20

LIST OF FIGURES – CONTINUED

Figure	Page
9. Figure 9 Confusion plot showing only the misclassified frames. Note that messages with index greater than 214 were predicted by the algorithm to be a message not present in the test dataset.	21
10. Figure 10 Simple diagram of various communication protocols.	24
11. Figure 11 Schematic of the Experiment Layout Completed at INL.	25
12. Figure 12 Example of mixed signals (A and B in the block diagram) and corresponding original waveforms (Ta and Tb in the block diagram). The goal of the LSTM is to reconstruct the original waveforms from the mixed signals.	28
13. Figure 13 Training and validation losses for the LSTM model. The model training behavior follows the ideal pattern of reducing validation loss (indicative of improving model performance) while keeping training and validation loss close (indicative of avoiding overfitting).	29
14. Figure 14 Actual and predicted values for the first target (top) and the second target (bottom). The actual values (Ta and Tb) are obtained from the oscilloscope using the active directional coupler (see Fig. 11). The predicted values are obtained by passively tapping the transmission lines at point A and point B, and feeding the passively obtained information into the trained LSTM. Each trace consisted of 500,000 training points, but the plots show only the first 650 samples.	30
15. Figure 15 Histogram of point-by-point errors between the actual target values (Ta and Tb) and the predicted target values (derived from feeding A and B into the LSTM). The prediction error has a mean of 0.00186 and a standard deviation of 0.102.	31

NOMENCLATURE

ADC: Analog to Digital Converter
AI: Artificial Intelligence
ARIMA: Autoregressive Integrated Moving Average
ATM: Asynchronous Transfer Mode
CD: Collision Detection
CNN: Convolutional Neural Network
CRC: Cyclic Redundancy Check
CSMA: Carrier Sense Multiple Access
DSP: Digital Signal Processing
EEE: Energy Efficient Ethernet
EEEP: ECOWAS Energy Efficiency Policy
GNN: Graph Neural Network
IEEE: Institute of Electrical and Electronics Engineers
IoT: Internet of Things
IP: Internet Protocol
IT: Information Technology
k-NN: k-Nearest Neighbors
LAN: Local Area Network
LSB: Least Significant Bit
LSTM: Long Short Term Memory
MAC: Medium Access Control
MAN: Metropolitan Area Network
ML: Machine Learning
MSB: Most Significant Bit
OSI: Open Systems Interconnection
OT: Operational Technology
RG-174: 50 ohm coaxial cable
RG-316: 50 ohm coaxial cable
RNN: Recurrent Neural Network
RTU: Remote Terminal Unit
SCADA: Supervisory Control and Data Acquisition
SLA: Service Level Agreement
SP: Service Provider
SVM: Support Vector Machine
TCP : Transmission Control Protocol
TSN: Time-Sensitive Networking
UDP: User Datagram Protocol
UNI: User Network Interface
WAN: Wide Area Network

ABSTRACT

This research explores the ability of machine learning to perform signal separation of an Ethernet style encoded, full-duplex communication. Typical signal separation currently requires an active tap of the communication line, followed by a recombination and retransmission of the data. The purpose of this research is to study a passive approach to the acquisition of data from a full-duplex signal. The machine learning model used in this research is a long-short-term memory recurrent neural network (LSTM-RNN), typically expressed as LSTM. The results show that the LSTM was largely successful in recreating the transmission signal from the measured data points, though the separated signals of the full-duplex Ethernet protocol have not yet been tested using a manual decoding method.

INTRODUCTION

General

Machine learning has become an essential tool in modern fields of science and industry, including electrical and computer engineering, physics, astronomy, healthcare, finance, and many other disciplines. From the creation of the first neural network using an electric circuit completed by Warren McCulloch and Walter Pitts in 1943, machine learning applications have surged in the 21st century. The success of machine learning has been a direct consequence of Moore's Law. As modern computers have increased in computing power, this has allowed for larger data sets to be used to train machine learning models as well as more sophisticated machine learning models to be developed [25]. This thesis will present research on using a Long Short Term Memory (LSTM) - Recurrent Neural Network (RNN) to study digital signal separation for a Gigabit Ethernet communication.

Ethernet Communication

A common form of Ethernet used today is the 100 Base-TX [40]. The 100 represents the transmission speed of the digital signal of 100 Mb/s. The transmission and receive of a signal for this Ethernet can be accomplished under two distinctions; half-duplex communication or full-duplex communication [3, 48, 54]. In full-duplex communication, signals propagate simultaneously between two endpoints on a single shared conductor. The signals are transmitted on the same carrier frequency. In half-duplex communication, two endpoints share a communication line, but only one node is allowed to drive the line at a given time. The introduction of full-duplex communication is made possible through a combination of increases in transmission speeds and advances in RF and baseband techniques [12].

Unfortunately, the extra bandwidth obtained via full-duplex communication comes at the cost of complications in signal separation [17]. Recent explorations in full-duplex communication channel signal separation include RF self-interference cancellation [19, 38], analog beamforming [7], and independent component analysis [11, 44]. Additionally, machine learning techniques have been successfully applied for signal separation in other application domains, including audio [21, 53], radar [29], and even protein analysis [30].

Machine Learning

Machine learning is a subfield of artificial intelligence. In machine learning, algorithms are used to attempt to allow computers to learn from large data sets, and then accomplish tasks, based on the studied data, without being explicitly programmed to do so. Two primary distinctions within machine learning are physics-based models and data-driven models. This paper focuses on a data-driven model, specifically the LSTM model.

LSTM is a type of RNN, which is a type of deep learning. Deep learning is a subfield of machine learning. Deep learning models are based on artificial neural networks that contain multiple layers, called deep neural networks [52]. As data is passed through the layers, each layer performs a mathematical operation on the data. Each layer serves as the input to the next layer. Ultimately, resulting in some output from the neural network [52]. This approach can be very useful with data sets that have difficult to define characteristics. A disadvantage of deep learning is it is computationally expensive and requires very large amounts of data to be successful [52].

An LSTM is a specific type of RNN. One of the challenges faced by RNN models when working with time series data is learning long term dependencies. In an RNN, data from distant time steps can be essential for making accurate predictions of the current state [52]. LSTM models eliminate this challenge by including a memory cell which can hold data from distant time steps [52].

Summary

In this work, we seek to evaluate the effectiveness of machine learning techniques applied to the problem of full-duplex signal separation. We focus our efforts on implementing a long short-term memory-recurrent neural network (LSTM-RNN) [45]. LSTM-RNNs have been effectively used in many time-series analysis tasks [23]. Our goal is to use an LSTM-RNN to potentially decode the mixed signal in a full-duplex communication system in order to recreate the original transmitted waveforms.

The layout of this thesis will begin with this introduction and follow with a background chapter. The background chapter will present research from other parties that studies machine learning, communication protocols, and research that has attempted to apply machine learning to communication protocols within various scientific and industrial disciplines. This thesis will then continue with two technical chapters summarizing research completed by Keith and colleagues from both Montana State University and Idaho National Lab. This thesis will finish with a conclusion section, discussing the success of the research presented as well as possible future considerations.

BACKGROUND

Machine Learning Literature Review

Since its first incarnation in the form of an electric circuit neural network in 1943, machine learning has become an integral tool to modern research in science and engineering [25]. This chapter will present examples of completed research with machine learning as a primary form of data analysis. The fields of study presented include genetics, electrical engineering, and astronomy. These three fields were chosen for being fields of interest to the author, but machine learning application is present in many other disciplines. While these studies discuss the use of general machine learning algorithms, this section will then present research using the LSTM model type.

Machine learning methods can be divided into supervised and unsupervised models. Supervised models are used on data which have labels associated with each data point, such as "gene a" or "non-gene" for example. These models, once trained, then attempt to predict similar labels for other data sets. Unsupervised models find patterns in data without labels [28]. Each of these methods has strengths depending on the anticipated outcome of the problem in question. Similarly, models can be classified as generative or discriminative models. Generative models output a statistical distribution over the input data and are most successful at interpretation of data. Discriminative methods only model labels and are typically used for prediction of data [28]. LSTM models are typically generative.

Libbrecht and Noble present research on the analysis of genome sequencing data sets. Their work includes annotation of sequence elements and epigenetic, proteomic, and metabolomic data. They present challenges and considerations in applying supervised, semi-supervised, and unsupervised learning methods to these data, as well as results using generative and discriminative models. Their research concludes with discussion on choosing between these methods and the application to genetic and genomic data set research [28].

Banik and colleagues present the following research on the prediction of electrical energy consumption based on data studied by machine learning techniques. Due to market deregulation and the introduction of renewable resources, the forecast of electricity demand has become increasingly relevant. Advanced intelligent models are built to provide precise power forecasts. The use of intelligent forecasting algorithms is a key feature of smart grids. They are also an effective tool for resolving uncertainty for better decisions about energy and cost efficiency, such as scheduling generations, ensuring system reliability and power optimization, and economic smart grid operations. Forecasting algorithms must have very high prediction accuracy as many important activities of power operators, such as load dispatch, depend upon this short-term forecast. This research completed by Banik et al proposes a model for the estimation of the consumption of electricity in Agartala, Tripura in India. This model can accurately predict the next 24 hour load with an estimation of load from 1 week to 1 month. Several characteristics in the city are analysed to extract variables that could affect the pattern of electricity consumption directly. This research provides results to significantly improve the accuracy of the prediction through ensemble machine learning processes. The research uses the models Random Forest, Extreme Gradient Boosting (XG Boost), and an ensemble of the two methods. The ensemble results in an improved accuracy of 15-29 percent. The results also provide information in connection with energy consumption [6].

Fluke and Jacobs present the following research on the application of machine learning and artificial intelligence (AI) in astronomy. Their review surveys contemporary published literature on machine learning and AI in astronomy and astrophysics. Various applications of machine learning are now firmly established in astronomy. Random forests, support vector machines, and neural networks are now having a genuine impact in applications as diverse as; discovering extrasolar planets, transient objects, quasars, and gravitationally lensed systems. Further areas of relevance include forecasting solar activity, and distinguishing between

signals and instrumental effects in gravitational wave astronomy. Machine learning and AI applications span seven main categories of activity; classification, regression, clustering, forecasting, generation, discovery, and development of new scientific insights [13].

Greff and colleagues present research on LSTM architecture. LSTM architecture for RNN was first proposed in 1995 and since then, several variants of LSTM have been studied. In recent years LSTM has become the state-of-the-art model for many machine learning problems. Greff and colleagues research presents a large scale analysis of eight LSTM variants on three different tasks; speech recognition, handwriting recognition, and polyphonic music modeling. The hyper-parameters of all LSTM variants for each task were optimized using random search, and their importance was assessed using the functional Analysis of Variance framework. The research summarized 5400 experimental runs. This was the largest study of LSTM architecture at the time of publishing. The results show none of the variants were significantly better than the standard LSTM architecture. They determined the forget gate and the output activation function are the most critical components of the LSTM architecture [18].

Communication Technology Literature Review

Ethernet is a protocol that allows computers to communicate with each other over wired networks through devices such as routers, switches, hubs, and wireless protocols. The Ethernet protocol connects LANs, WANs, Internet, IoT devices, cloud, and Wi-Fi systems into a single global communications network. Ethernet is one of the original networking technologies and was invented in 1973 by a team at the Xerox Corporation. Ethernet remains the dominant wired networking technology due to its simplicity, adaptability, and backward compatibility. As AI workloads have begun to increase in modern industry, engineers have focused on ensuring that Ethernet technology will be able to keep up with this new high-performance networking demand [8].

One of the successes of Ethernet is the high efficiency of the protocol. Ethernet breaks down information into "frames" that contain the source and destination addresses of the information being transmitted. This enables efficient routing across the network. Older Ethernet technology uses collision detection techniques and modern Ethernet uses full-duplex communication, which eliminates any possible collision between transmitting and receiving communication signals [8]. This full-duplex communication is a primary point of interest in the research presented in Technical Chapters 1 and 2. A wealth of research on traditional (non AI driven) Ethernet communication applications has been completed. Examples of Ethernet communication used in industry are presented here.

J.D. Decotignie presents an article on the use of Ethernet-based real-time industrial communications. This paper reports that despite early attempt to use Ethernet in the industrial context, only recently (as of the early 2000's) has the technology gained attention as a support for industrial communication. A number of vendors are offering industrial communication products based on Ethernet and TCP/IP as a means to interconnect field devices to the first level of automation. Others restrict their offer to communication between automation devices such as programmable logic controllers and provide integration means to existing fieldbuses. This paper details the requirements that an industrial network has to fulfill. It then shows how Ethernet has been enhanced to comply with the real-time requirements in particular in the industrial context. Lastly, the paper presents how the requirements that cannot be fulfilled at layer 2 of the OSI model can be addressed in the higher layers adding functionality to existing standard protocols [55].

Danielis and colleagues completed the following research on companies in the automation industry. Their research provides a summary of different industrial Ethernet protocols for real-time data transmission via Ethernet in automation environments. For companies in the automation industry, the development of real-time Ethernet to connect devices is of high economic interest. The goal is to replace conventional fieldbus systems. This is a

challenging task. The original Ethernet standard IEEE 802.3 was not designed for real-time data transmission. Also, protocols like TCP, UDP, and IP do not consider real-time requirements. Therefore, adaptations on several OSI layers becomes necessary to make an industrial system meet these requirements. To accomplish this, a variety of realtime capable industrial Ethernet systems have been developed, thus solving the problems of standard Ethernet, TCP/IP, and UDP/IP based communications. Danielis and colleagues research provides results analyzing the advantages and disadvantages of these protocols with regard to their sustainability in terms of their realtime capabilities, reliability, scalability, self-configuration of the network, and hardware requirements. The authors discuss consequences for future developments based on the number of connected devices growing over time and the growing computational power of the "industrial internet" [14].

Zier and colleagues present the following research on the trade-offs of using Ethernet based communication in public communication services. Their study discusses how enterprises and residential customers increasingly take advantage of more sophisticated applications and evolve their business models. At the same time, service providers (SP) see a declining revenue from the existing connectivity services. In response to this trend, SPs begin to offer highly customized, high-bandwidth network services that complement and interwork with the existing leased line, ATM, or frame relay offerings. Ethernet is quickly becoming the customer UNI of choice. Ethernet is a packet based technology, and complements IP-based services such as content, voice, data, video, and additional value-added services over a high-speed access connection. Ethernet also provides a flexible high-speed connection to the SP network and eliminates access bottlenecks. Using Ethernet as the common access interface allows providers to employ flexible transport services that enable additional value-added services at layer 3 (of the network layers) and above. Also, the rollout of Ethernet services challenges the installed base, given that bandwidth per customer is usually much higher and additional attributes for tight SLAs apply. These challenges suggest an updated

approach to network deployment and rollout is needed. An updated approach in which networks need to be planned and built in a service-centric fashion. The article written by Zier et al begins with a look at the driving forces of Ethernet-based public communication services. The article identifies five generic services for Ethernet MANs/WANs and discusses related deployment issues of the different service options. These options include the degree of customization, geographic reach, bandwidth profile of the service, and evolution aspects of the installed base [55].

Machine Learning Applications in Communication Technology Literature Review

The problem of developing a real-time Ethernet network is one of many that engineers are attempting to solve with the application of machine learning algorithms. This section will present several examples of completed research on machine learning applications to Ethernet communication protocols. This section concludes the Background chapter for this thesis.

Mai and colleagues present the following research to explore how machine learning may provide a viable alternative to conventional scheduling analysis to determine if a real-time Ethernet network meets a set of timing constraints. Mai and others applied a supervised machine learning technique called k-Nearest Neighbors (k-NN). This algorithm is used here to compare its accuracy and running times against precise scheduling analyses developed in Network-Calculus. The experiments consider different TSN scheduling solutions based on priority levels and traffic shaping. The results are obtained for an automotive network topology. These results suggest that k-NN is efficient at predicting the feasibility of realistic TSN networks, with an accuracy range of 91.8 percent to 95.9 percent. These results depend on the exact TSN scheduling mechanism and a speedup of 190 over a schedulability analysis for 10^6 considerations. One trade-off discovered with this approach is, machine learning produces a certain number of "false positive" results. There is evidence to suggest a number of false positive results may be tolerable in this area of study [32].

Lucas Torlay presents his completed master’s thesis from Clemson University with the following research. The cybersecurity of power systems is vulnerable to the threat of attacks due to a lack of physical layer device authentication techniques for OT communication networks. OT networks cannot support the active probing cybersecurity methods that are popular in IT networks. In addition, both active and passive scanning techniques are susceptible to medium access control (MAC) address spoofing when operating at Layer 2 of the Open Systems Interconnection (OSI) model. Torlay’s thesis analyzes the role of deep learning in passively authenticating Ethernet devices by their communication signals. This method operates at the physical layer, or Layer 1, of the OSI model. The security model collects signal data from Ethernet device transmissions, applies deep learning to gather distinguishing features from signal data, and uses these features to make an authentication decision on the Ethernet devices. This proposed approach is passive, automatic, and spoof resistant. The role of deep learning is determined to be critical to this security model [51].

Mai and Navet present research exploring improvements to deep-learning methods used as prediction to switched network Ethernet configurations. The machine learning model used in this research is a graph neural network (GNN). This is an advanced machine learning model that has recently been applied to encode Ethernet configurations as graphs and predict their feasibility in terms of meeting deadline constraints. GNN models have proven to be robust to changes in the topology and traffic patterns with respect to the training set. However, the moderate prediction accuracy of the model, 79.3 percent at the lowest, hinders the application of GNN real-world problems. This study proposes improvements to the base GNN model in the construction of the training set and the structure of the model itself. First, new training sets are introduced to the GNN model. These sets are more diverse in terms of topologies and traffic patterns and focus on configurations that are difficult to predict. The GNN model is then enhanced with more powerful activation functions, multiple channels, and implementation of a technique called global pooling. The prediction accuracy of the

GNN with a combination of the suggested improvements increases significantly, up to 11.9 percent on the same 13 tests. An important determination from this research is that these improvements only marginally increase the time it takes to predict unseen configurations. The speedup factor is still from 50 to 1125 compared to schedulability analysis, which allows a far more extensive exploration of the design space [31].

Segolene et al. present research studying the application of machine learning to improve the prediction-based Energy Efficient Ethernet (EEE) strategy. According to their paper, Ethernet consumes maximum energy even when there is no data transmission. To reduce the power consumption, IEEE 802.3az standardizes the Energy Efficient Ethernet that enhances Ethernet with the low power idle state without data transmission. However, this standard does not describe the specific strategy about when the Ethernet link will enter or exit the low power idle state. The EEEP strategy was proposed for the 1-10 Gbps EEE to reduce power consumption. Specifically, EEEP predicts the future traffic in a time window by the Autoregressive Integrated Moving Average (ARIMA) model and determines when to enter or exit the low power idle state according to the prediction results. However, the EEEP strategy relies on the prediction accuracy of the ARIMA model for good energy saving. This paper proposes to use the Long Short Term Memory (LSTM) model for EEEP to improve the prediction accuracy. Due to the historic traffic information, the LSTM model can achieve about 11 percent improvement on the accuracy compared to ARIMA, and thus helps EEEP to achieve better energy saving, according to the author’s trace-driven simulation results [43].

SUPERVISED MACHINE LEARNING FOR MODBUS COMMUNICATION PROTOCOL DECODING

The majority of the technical content for this chapter has been accepted for publication in the 2025 Intermountain Engineering, Technology and Computing conference: Skyler Reid, Maximus Marceau, Keith M. Filler, Keith D. Mecham, and Bradley M. Whitaker, "Supervised Machine Learning for Modbus Communication Protocol Decoding," IETC 2025.

Introduction

This study explores machine learning for decoding Modbus RTU data using K-Nearest Neighbors (KNN) models. An initial KNN model trained on 8,000 packets achieved 95.15% accuracy. Although ML improves generalization, accuracy still falls short of deterministic methods. These findings have implications for Modbus traffic analysis, intrusion detection in industrial networks, and adaptive error correction in real-time monitoring systems. By refining ML-based decoding, future work could enable more efficient anomaly detection and predictive maintenance in industrial automation and cybersecurity applications.

Modbus RTU is a fundamental serial communication protocol used for many basic communication tasks [49]. With a maximum baud rate of 115,200, data is transmitted in 8-bit packets. Each packet also includes a start bit, a checksum bit, and an end bit, forming an 11-bit unit. These packets are grouped into data frames, typically spanning between 1322 to 1340 packets in length. Encoding and decoding these packets is well understood and is common practice in many computing applications [2, 15]. Many other communication protocols exist as well, each with their own encoding and corresponding decoding schemes [1, 5, 24, 46, 50].

While communication protocols are straightforward when the encoding scheme is known, it can be very difficult to reverse engineer an encoding scheme in order to create

a decoding scheme by analyzing only the transmitted data [9]. In this work, we apply machine learning to develop a decoding algorithm beginning with the knowledge of the pre-encoded data and the received encoded signal, but without initial knowledge of the encoding algorithm.

As a widely adopted industrial communication protocol, Modbus has been the subject of extensive research. However, the application of machine learning to Modbus signal decoding remains relatively unexplored. Instead, much of the existing literature focuses on leveraging machine learning techniques for anomaly detection and cybersecurity applications [4, 39]. Prior studies have demonstrated that models such as extreme gradient boosting and convolutional neural networks (CNNs) can detect faults and cyberattacks in Modbus networks with near-perfect accuracy, reaching values as high as 0.9999 [10, 37, 41]. Some of the same machine learning techniques explored in this study, including long short-term memory (LSTM) networks and support vectors machines (SVMs), have also been applied in these anomaly detection frameworks.

Beyond security applications, research has also explored the potential for embedding hidden communication channels within Modbus transmissions [27]. Several plausible covert channels have been identified, demonstrating the feasibility of transmitting additional data within standard Modbus traffic. However, further investigation is required to assess the scalability and robustness of this approach in real-world deployments.

Another key area of research in Modbus protocol analysis is the implementation of Hamming codes for error detection and correction [22, 42]. These techniques enable the identification of single- and double-bit errors, with automatic correction of single-bit errors, enhancing the reliability of Modbus-based communication systems. The integration of such error-correcting mechanisms is particularly relevant in environments where transmission integrity is critical.

The simplicity and relative security of Modbus communication makes it an ideal

candidate for developing a proof-of-concept decoder using machine learning techniques. We show that machine learning is effective at decoding the Modbus communication channel, given a collection of input-output pairs.

Methods

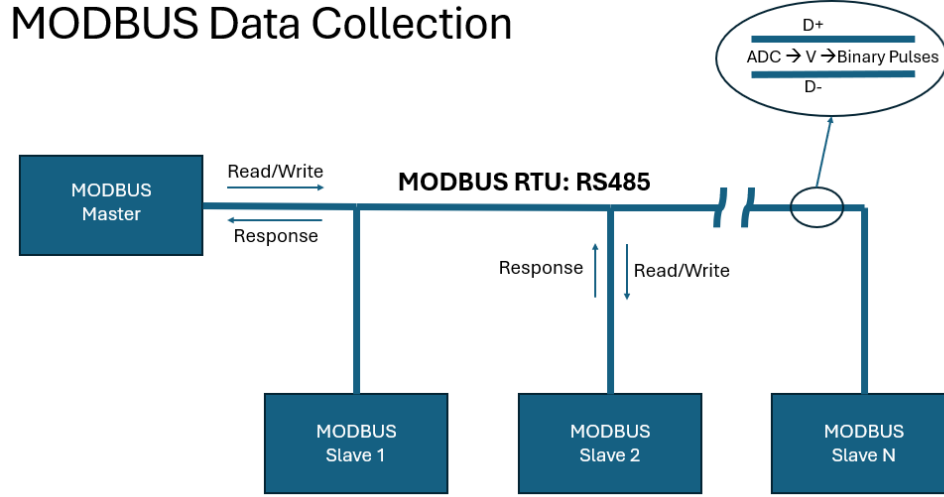


Figure 1: Experiment layout for the initial voltage measurements from a MODBUS communication scheme. These measurements are then converted into usable data for training the machine learning models.

Data Collection Figure 1 shows an example layout for an experiment in which initial signal voltage measurements were performed and recorded as data. Figure 2 displays an outline of the data creation process used in this study. The data creation process involved alternating transmissions (Tx) and responses (Rx) between two communication nodes, while constantly measuring the voltage level on the transmission line. There is a brief pause (transmitting only zeros) between corresponding Tx/Rx signals, and a longer pause after the complete Tx/Rx pair. This transmit/receive procedure was repeated 88,485 times, at which point one billion samples were collected by the oscilloscope. The one billion samples include information about 176,970 traces plus the zero-only pauses between traces. The actual transmitted data bits for each trace were stored in a log for comparison.

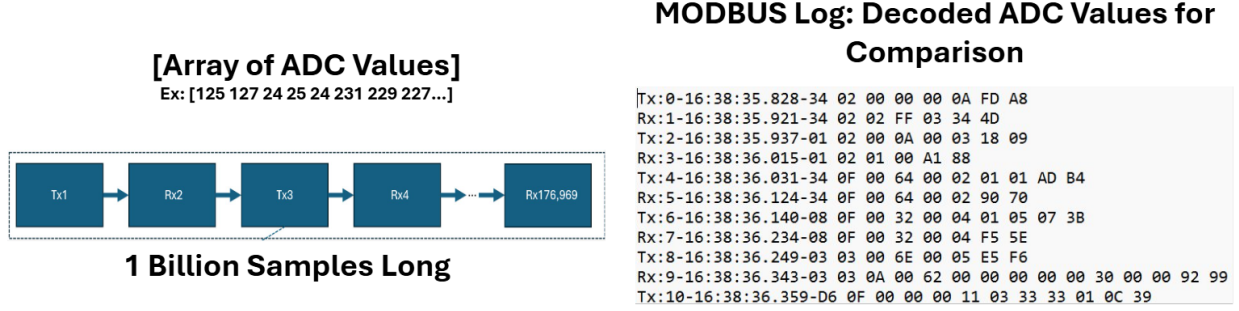


Figure 2: Illustration of the process of converting data frames to modbus log entries. The data creation process involved alternating transmissions (Tx) and responses (Rx) between two communication nodes. There is a brief pause (transmitting only zeros) between corresponding Tx/Rx signals, and a longer pause after the complete Tx/Rx pair. This transmit/receive procedure was repeated 88,485 times, at which point one billion samples were collected by the oscilloscope. The one billion samples include information about 176,970 traces plus the zero-only pauses between traces.

Each trace consists of an address packet, a function code packet, and between 7 and 15 data packets. In turn, each packet consists of a start bit, an 8-bit hex value for transmission, a parity bit, and a stop bit, for a total of 11 bits. Lastly, the oscilloscope was set to sample the voltage at 5 Gsps, which corresponds to about 10 samples per bit. Fig. 3 shows plots of a trace, identifying the first three packets within the trace (the address packet, the function code packet, and the first data packet).

Prior to training, the data were preprocessed to ensure uniformity in frame length. Frames shorter than the mean length were zero-padded, while longer frames were downsampled to match the mean length. Frames consistently had around 1,330 samples with minimal variation; in fact, 50% of all the packets were between 1122 and 1130 samples long. Depending on which subset of packets were analyzed, the mean length was between 1129 and 1135 samples long.

For the purposes of this study, the zero-padded or downsampled waveform is considered to be the input that needs to be decoded. The two-character hexadecimal (hex) value associated with the packet (ignoring the start, parity, and stop bits) is taken from the log

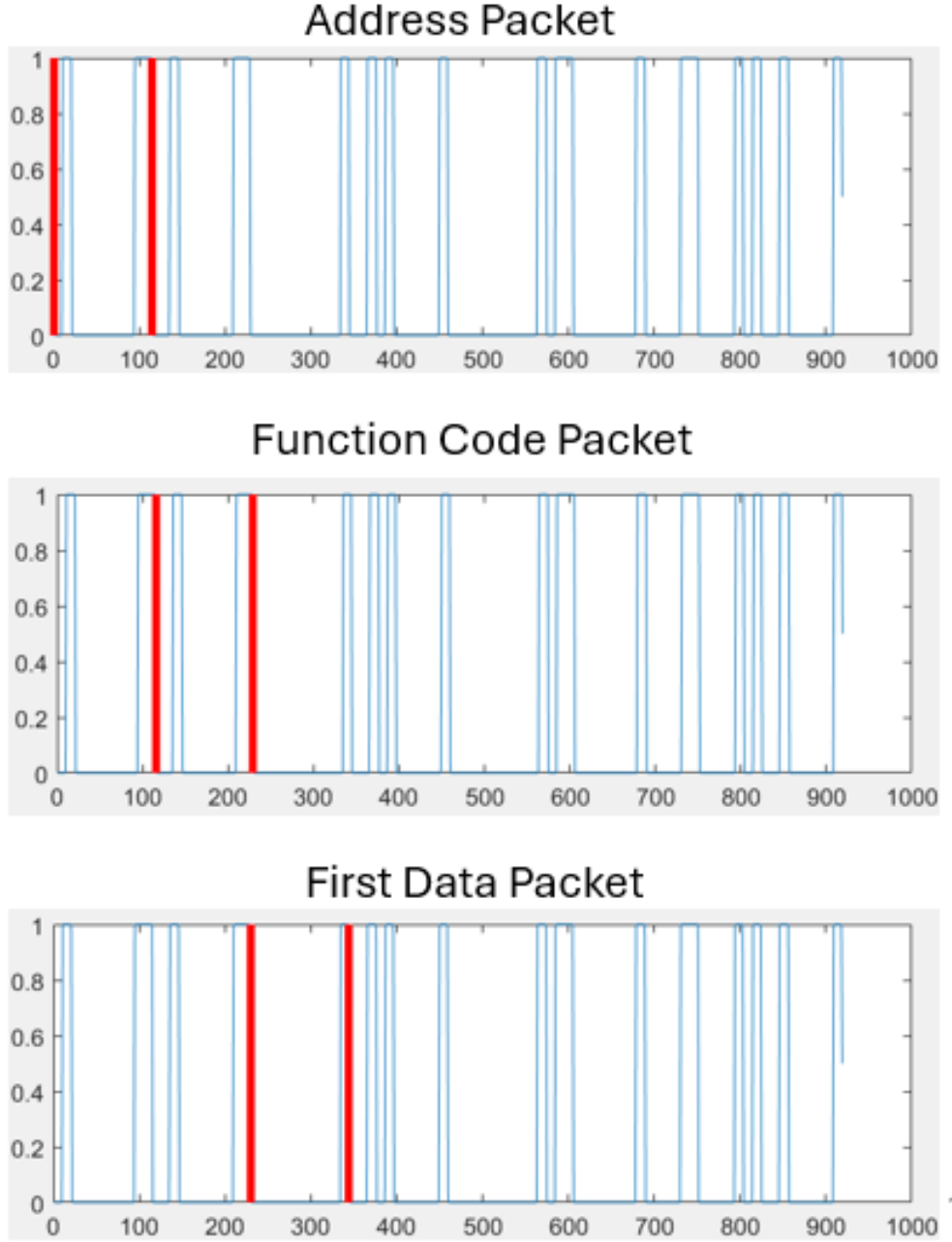


Figure 3: Trace structure showing address packet, function code packet, and the first data packet within the trace. Note that each packet contains 11 bits and about 110 samples, meaning that the signal has approximately 10 samples per bit.

file to be the ground truth, or target, for the decoder. For this work, we develop two decoding algorithms: one is developed manually based on the Modbus encoding scheme, and the other

is developed using supervised machine learning.

Manual Decoder Initially, we developed custom decoding algorithms to manually parse and decode each frame, in order to provide ground-truth data for training machine learning models. To identify individual data frames within a complete transmission or response (referred to as a trace), we leveraged the characteristic separation of frames by extended sequences of zeros, making the segmentation process straightforward.

Figure 4 shows the data structure of a typical MODBUS packet. Each data packet contains 11 bits, each of which is approximately 10 samples long. However, the bits could be as long as 14 samples or as short as 7 samples. To account for these possibilities, our decoding algorithm counted the number of samples in three high-to-low bit transitions, then divided that number by three and used the result as the length of one bit. This ensured that the correct bit length was used, and error did not compound with each data packet.

Once a binary sequence could be determined by the algorithm, the sequence was flipped to account for little-endian encoding and converted to hexadecimal (hex) format. This sequence of hex values was then compared to the values stored in the MODBUS log to determine the accuracy of the created algorithm. An example of this procedure is shown in Figure 5.

Machine Learning Decoder As discussed previously, the dataset used for machine learning comprised 176,000 frames. Preprocessing involved normalizing frame lengths:

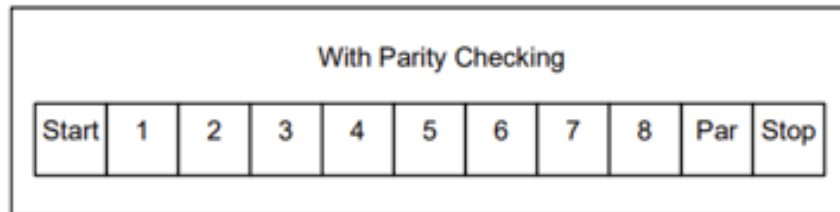
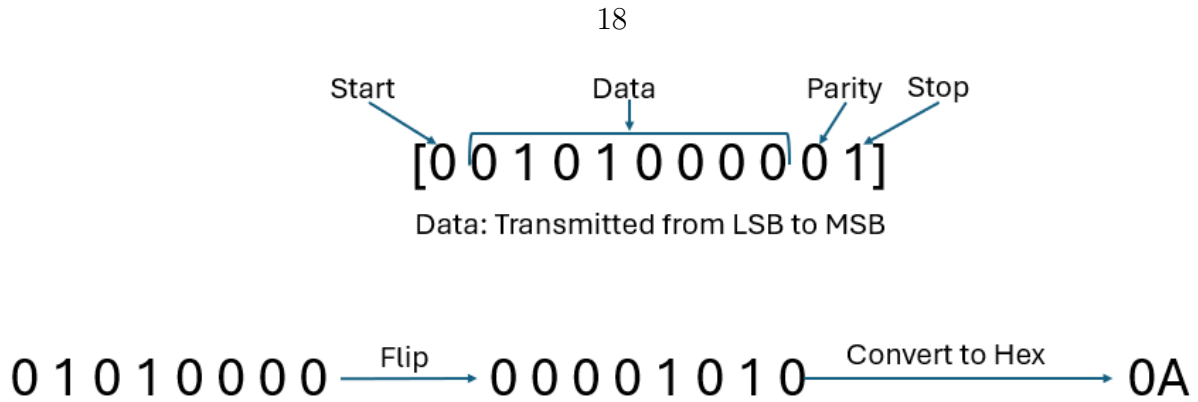


Figure 4: Data structure of a MODBUS packet.



After looping through and decoding each packet

Final Decoded Message: 01 02 00 0A 00 03 18 09

Figure 5: An example of analyzing a determined binary sequence. The first step in this procedure is to account for the little endian style encoding scheme typical to MODBUS data. This requires the initially determined binary sequence to be flipped from LSB first to MSB first. This new binary sequence is then converted to the hexadecimal equivalent and compared to the MODBUS log data.

shorter frames were zero-padded, and longer frames were downsampled to the mean length. This preprocessing step ensured consistent input dimensions for machine learning models while preserving data integrity.

For computational efficiency, we reduced the training set size to 10,000 packets while maintaining diversity, enabling more accurate classification of unseen data. The 10,000 frames include 588 unique messages. Of these, 468 are represented only once in the dataset.

For the machine learning classification task, we explored four basic machine learning models: Decision Trees, Ensemble Trees, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). Preliminary results indicated that the Tree methods had poor performance. In addition, while both SVM and KNN had similar performance, the KNN was much faster. Based on this, the classification algorithm selected for this project was the KNN algorithm.

TABLE I
HYPERPARAMETER SEARCH SPACE FOR KNN

Hyperparameter name	Search space
Number of Neighbors	[1, 8000]
Normalization	{true, false}
Distance Metric	{Euclidean, City Block, Mahalanobis, Minkowski, Cosine, Correlation, Hamming, Chebyshev, Spearman, Jaccard}

Figure 6: Table of hyperparameters for KNN training procedure.

Table 1 in Figure 6 shows the hyperparameters that were explored during the training process for the KNN classifier.

Results

Manual Decoder The manual approach was developed based on a priori knowledge of the Modbus encoding scheme. As a result, it achieved 100% accuracy in splitting frames into packets and decoding individual packets. Figure 7 shows the data values determined from the decoding algorithm compared with the recorded MODBUS data log values.

Frame	Decoded Hex Values	MODBUS Log Hex Values	Match (Yes/No)
Tx1	34 02 00 00 00 0A FD A8	34 02 00 00 00 0A FD A8	Yes
Rx2	34 02 02 FF 03 34 4D	34 02 02 FF 03 34 4D	Yes
Tx3	01 02 00 0A 00 03 18 09	01 02 00 0A 00 03 18 09	Yes
Rx4	01 02 01 00 A1 88	01 02 01 00 A1 88	Yes
Tx5	34 0F 00 64 00 02 01 01 AD B4	34 0F 00 64 00 02 01 01 AD B4	Yes

Figure 7: Comparison of the determined hexadecimal values to the recorded MODBUS data log values. The results show a 100 percent accuracy in the manual decoding algorithm.

Machine Learning Decoder A K-Nearest Neighbors (KNN) model was initially trained on 8,000 frames, and the most effective hyperparameteres were determined to be: 1 nearest neighbor, no standardization, and City Block distance. When applied to the 2,000 frames

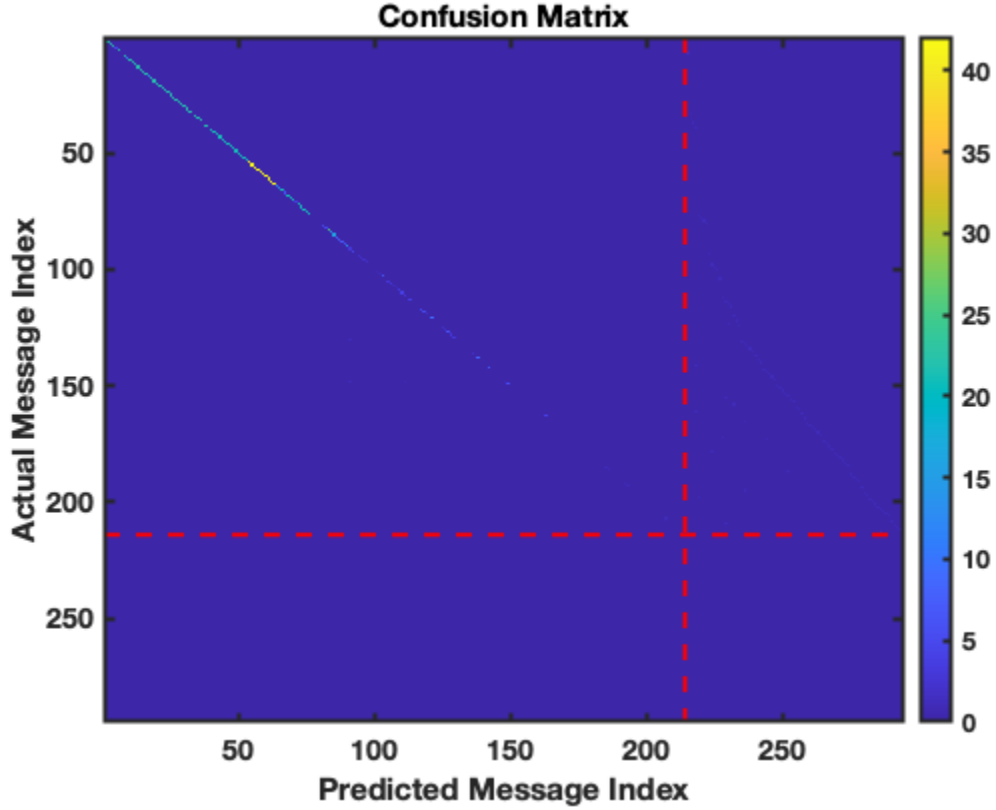


Figure 8: Confusion matrix. Note that messages with index greater than 214 were predicted by the algorithm to be a message not present in the test dataset. The presence of a strong diagonal indicates a successful classifier.

reserved for testing, the classifier achieved 95.15% accuracy. Note that the behavior of the model remained unchanged as it was scaled to be trained on 80,000 frames and tested on 20,000 frames. For brevity, we report only the detailed results on the smaller dataset. A confusion matrix for the test data is reported in Fig. 8.

The model’s performance plateaued because it relied on pattern recognition rather than true decoding. If a message was only present once in the dataset, it could belong either to the training set or the test set. If it was included in the test set, it was impossible for the KNN algorithm to select it as a possible solution, and was automatically misclassified.

For our test, the 95.15% accuracy includes correctly classifying 1903 of the 2000 frames. However, of the 97 misclassified data points, 94 of them were frames that were present only

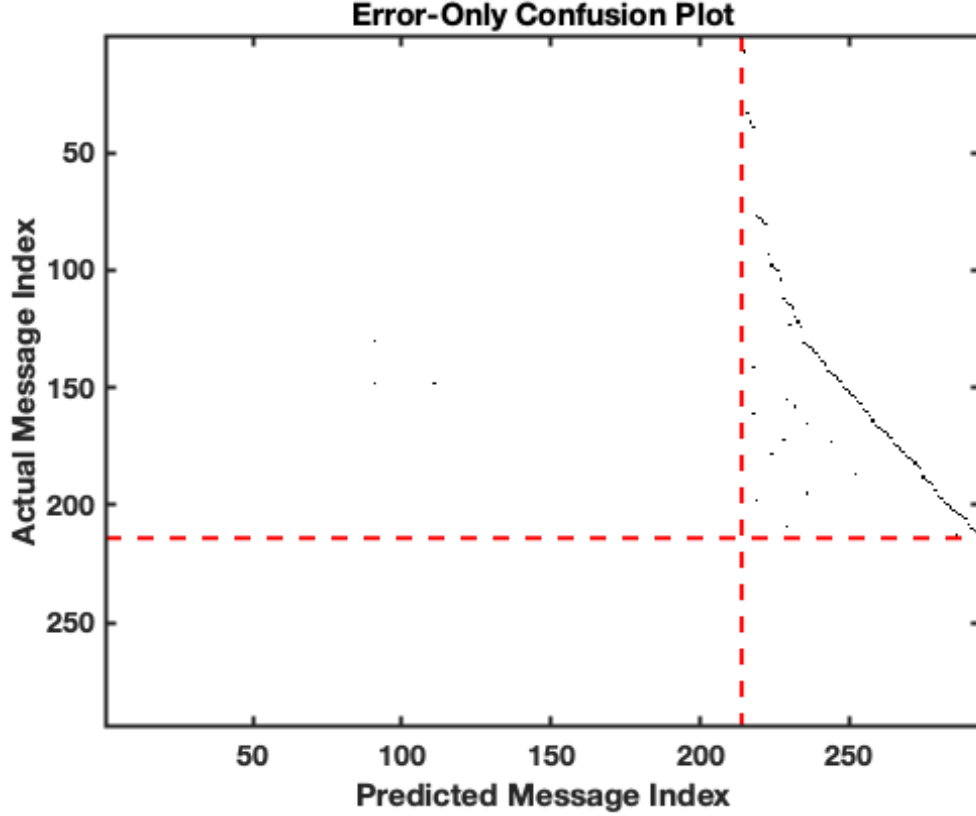


Figure 9: Confusion plot showing only the misclassified frames. Note that messages with index greater than 214 were predicted by the algorithm to be a message not present in the test dataset.

in the testing data. Thus, it was impossible for the algorithm to correctly label those 94 points. Instead, the algorithm predicted labels for these points that were not part of the test dataset. This behavior is illustrated in Figure 9.

Figure 9 shows a scatter plot of all misclassified points. The full dataset contained 588 unique messages. Of these, 494 were present in the training data, meaning that 94 messages were *not* present in the training set. The test set consisted of 214 unique messages. Of these, 94 are unique to the test set and 120 have some overlap with the training set. The top-left section of the figure shows the algorithm’s confusion when considering these 214 labels. There are only three misclassified frames, represented as black dots. In contrast, the top-right section of the figure shows that 94 incorrectly classified messages were predicted as

messages that were not even in the test set, because the predicted message number is greater than 214.

Ignoring the 94 points that are impossible to predict with machine learning classification results in an accuracy of 1903/1906, or 99.84%.

Conclusion

The deterministic decoding script achieved 100% accuracy, highlighting a fundamental limitation of machine learning in this application. Unlike deterministic decoding, the ML model could not guarantee correctness across all frames. Confidence interval analysis confirmed that even incorrect predictions had high confidence scores, limiting improvements through confidence-based filtering.

Despite the limitations of machine learning-based decoding, achieving a 99.84% accuracy on previously seen data is quite impressive. This approach may be extended to other communication protocols, such as 100 Base-TX and 100 Base-T1 Ethernet, to discover ways to create an automated decoder without a priori knowledge of an encoding scheme.

In summary, We have demonstrated the viability of machine-learning assisted decoding procedures for Modbus RTU communications. However, we have also demonstrated that such a model has an upper bound on accuracy, and will never be able to predict with 100 percent accuracy across a sufficiently large dataset.

LSTM FOR DIGITAL SIGNAL SEPARATION

The majority of the technical content for this chapter has been accepted for publication in the 2025 Intermountain Engineering, Technology and Computing conference: Keith M. Filler, Maximus Marceau, Skyler Reid, Stephen P. Ude, Noah J. Netz, Keith D. Mecham, and Bradley M. Whitaker, “Full-Duplex Gigabit Ethernet Signal Separation Using LSTM,” IETC 2025.

Introduction

This chapter will present the research completed for the application of an LSTM model to the passive data acquisition and analysis of a simulated communication signal. The intent of this research is to apply these methods to the passive data acquisition of a full-duplex Gigabit Ethernet communication. This chapter will cover the method of initial data acquisition from an experimental signal transmission, originally completed by Noah Netz and Stephen Ude while performing a research internship with Idaho National Laboratory [36]. This chapter will then present the approach used to apply an LSTM model to these data and present the observed results. The original LSTM experiment was completed by Netz and Ude. K. Filler and colleagues at MSU then used their results to recreate the LSTM experiment and study the results further. This chapter will conclude with a discussion on the interpretation and future research possibility determined from the experiment.

Methods

Figure 10 shows a simple diagram of three communication protocols. The protocol of interest with this research is the full-duplex communication. Typical communication protocols operate with a transmitter and a receiver. The full-duplex communication is capable of transmitting and receiving signals simultaneously. An example of this style of

communication is a telephone call, where parties on either end of the call can talk or listen simultaneously with each other [20].

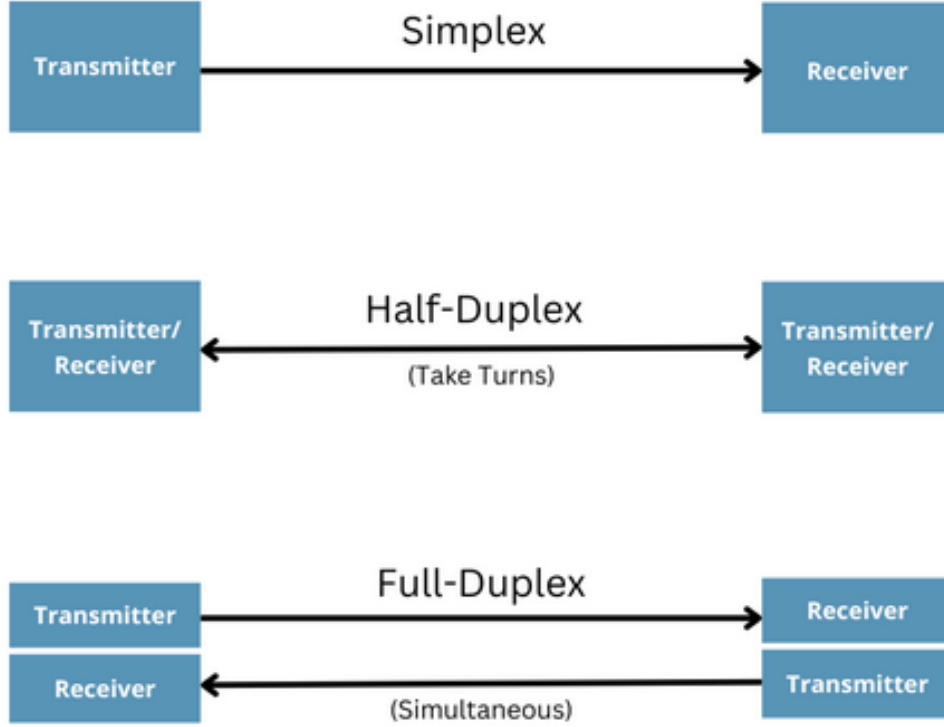


Figure 10: Simple diagram of various communication protocols.

To begin the study of full-duplex communication signals, a number of sample signals are collected and formatted to be used as data. This section begins with the details regarding the initial data acquisition from simulated communication signals. Following the data collection methods, the information for the application of an LSTM model for the analysis of the signal data is presented.

Data Collection The experimental setup is described in Figure 11. To begin the LSTM study, researchers at Idaho National Lab (INL) used a LeCroy TF-AUTO-ENET directional splitter to separate a 100 BASE-T1 signal into component signals called Ta and Tb. This initial approach was completed using an oscilloscope to sample a full duplex signal at two

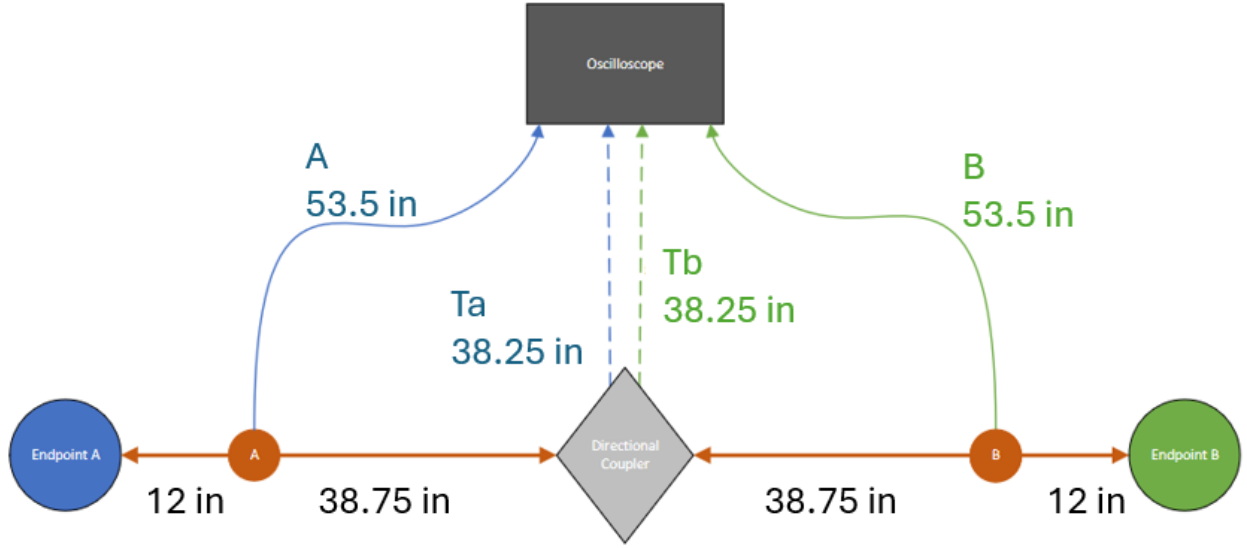


Figure 11: Schematic of the Experiment Layout Completed at INL.

points A and B on either side of the TF-AUTO-ENET directional splitter. This is an active procedure that includes decoupling the transmit and receive signals first, and then re-coupling the full-duplex communication signal. For this experiment, researchers also have direct passive line types at node A.

The labeled schematic in Figure 11 includes internal directional coupler traces and the lengths of the probes plus the cable lengths. T_a , T_b , the signal cables between endpoint A, the directional coupler, and endpoint B are all RG-316. These cables have a propagation velocity of $0.69c$, where c represents the physical constant designated for the speed of light in vacuum. Probes A and B appear to be made with RG-174, which has a propagation velocity of $0.66c$.

Using the successful LSTM experiment completed by the INL research team, the MSU research team was able to use the same data set and recreate the experiment in an effort to continue to study the results. The MSU researchers implemented the LSTM model in a Jupyter Lab environment written in the Python programming language. The model training

process was programmed to run for 50 epochs to allow for higher accuracy of the model. An epoch in machine learning indicates one complete pass of the training data set. The benefit of running multiple epochs during a training procedure is the model can improve over each epoch by adjusting internal parameters to better assess the data set [26].

LSTM for Signal Separation LSTMs are a commonly used machine learning tool for time-series analysis. These models are especially accurate in time-series forecasting, and have found application in financial forecasting, among other disciplines [47]. This type of model is a promising approach for the research goals of full-duplex signal separation because the signals of the full-duplex communication are time dependent. In fact, transmission signals, in general, are time dependent. This time dependence is critical to the success of a transmission signal in conveying information in the form of data. Transmission signals are characterized by parameters such as phase, amplitude, and frequency that change over time [16]. In this experiment, the goal of the LSTM is to recover the signal transmitted by measurement points A and B, which we call T_a and T_b , respectively.

The LSTM architecture used here begins by creating datasets from both the mixed (non-separated full-duplex) and original signals. The length of the data set indicates the number of mixed signal segments. These sets are converted into two-dimensional tensors for input into the LSTM. The tensor constructed from the original signals creates the target tensor from which the LSTM will learn the desired prediction [35].

After an initial dataset is loaded into the LSTM, the data is split into training and validation sets. These sets are also first converted into arrays, then two-dimensional tensors to fit the format of the LSTM. The number of features in the data is determined by the LSTM, and the training model is then instantiated with the specified parameters based on the shape of the data. The LSTM used here has a built in optimization protocol. This protocol allowed the model training to run for 50 epochs followed by data reporting on the

validation loss and validation R^2 for each epoch [35]. These results are summarized in Figure 13.

Figure 12 shows an example of an input waveform, labeled as a mixed signal, and the re-created output waveform, labeled as original waveform. The term "mixed signal" corresponds to the full-duplex communication signals measured at points A and B in the experiment layout schematic. The term "original signal" corresponds to the separated signals recorded at points Ta and Tb from the same schematic. The points Ta and Tb represent the communication signal after it has passed through the directional coupler. Note that the signals are normalized to have magnitude between -1 and 1. Traditional methods for tapping a communication channel require an active couple to provide Ta and Tb. Again, the goal of using an LSTM is to attempt to passively recover the active element of Ta and Tb.

Recovering a transmission signal is a challenge due to the simultaneous transmission and reception of the full-duplex communication. It is difficult to understand which signal is a transmission when both the transmit and receive are mixed together. The purpose of attempting to reconstruct signals Ta and Tb from the input communication data is to determine if an LSTM can take these mixed transmit and receive signals as input and then output two unmixed, or separated, signals. Success in doing so would suggest an LSTM model may be a good candidate for a passive procedure to acquire the individual transmit and receive data from a full-duplex communication.

Results

The training and validation losses correspond to the degree to which the training ML model agrees with the training and validation data sets. The training and validation losses shown in Figure 13 suggest the successful fit of the LSTM model to the respective data sets. The numerical values are measured after each epoch [34]. After allowing the LSTM training process to run for 50 epochs, the results yield a final validation loss = 0.0107 and a validation

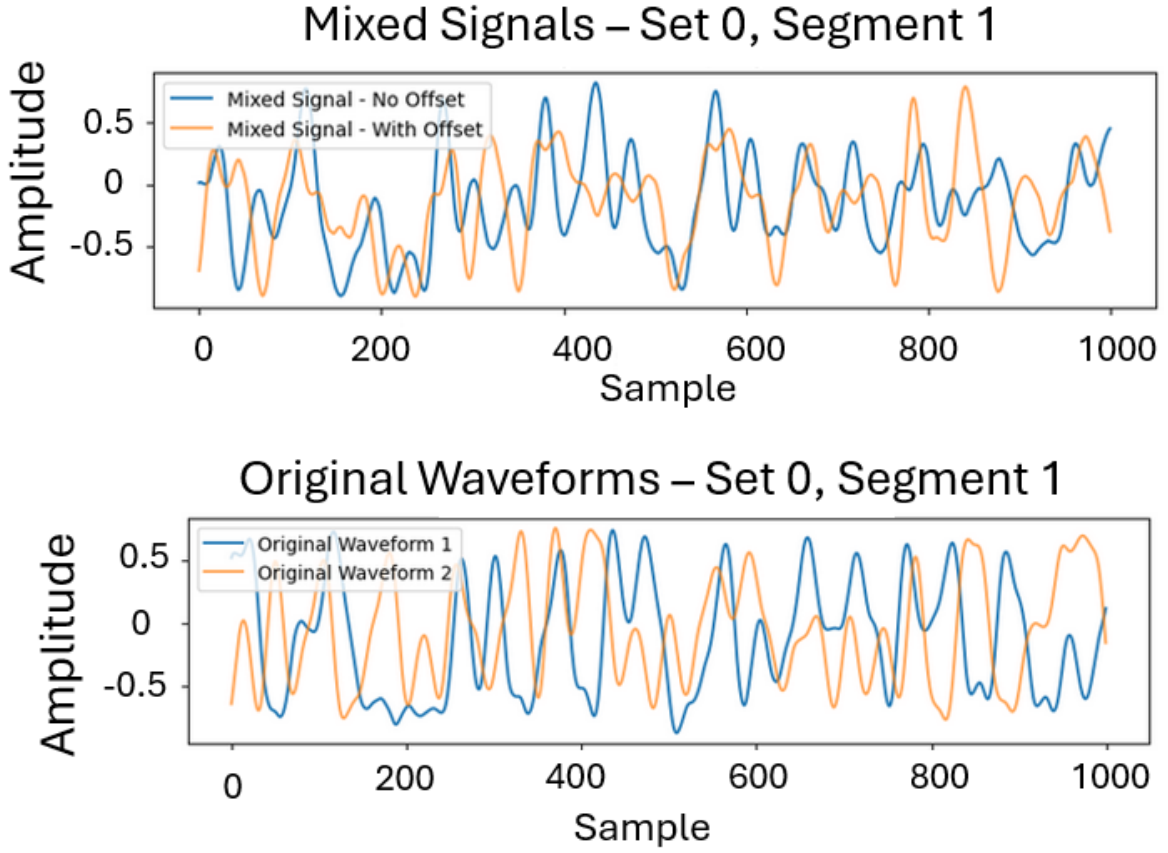


Figure 12: Example of mixed signals (A and B in the block diagram) and corresponding original waveforms (Ta and Tb in the block diagram). The goal of the LSTM is to reconstruct the original waveforms from the mixed signals.

R^2 of 0.9536.

Figure 14 shows the actual and predicted values for the first target (top) and the second target (bottom). The actual values (Ta and Tb) are obtained from the oscilloscope using the active directional coupler (see Fig. 11). The predicted values are the output result from the completed LSTM program. These waveforms are obtained by passively tapping the transmission lines at point A and point B, and feeding the passively obtained information into the trained LSTM. Each trace consisted of 500,000 training points, but the plots show only the first 650 samples.

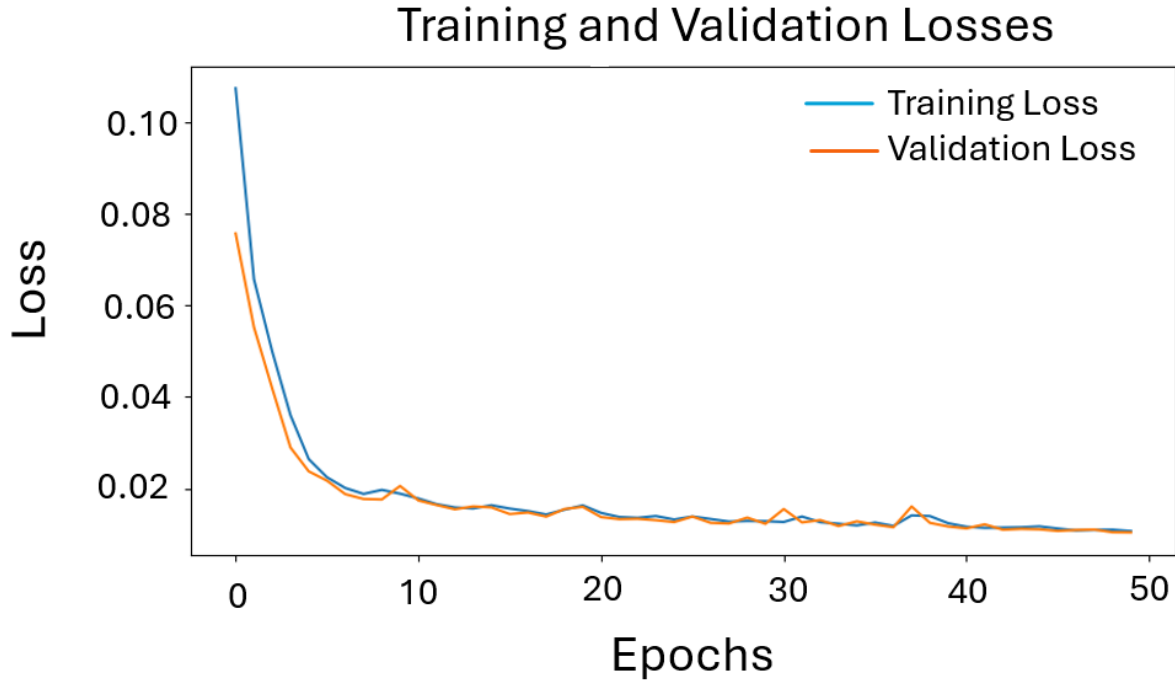


Figure 13: Training and validation losses for the LSTM model. The model training behavior follows the ideal pattern of reducing validation loss (indicative of improving model performance) while keeping training and validation loss close (indicative of avoiding overfitting).

Figure 15 shows the histogram of prediction errors of the LSTM model. A histogram of prediction errors is a useful tool for determining the accuracy of a machine learning model. On the x axis the histogram displays the range of values for variations between actual (target) values when compared with the value determined by the ML model. The y axis shows the frequency of the error for each particular value determination. A prediction error of 0 indicates a perfect correlation between the target value and the ML determined value.

The prediction error results clustered around zero tell researchers that this model is a good fit to the data. The prediction error follows a Gaussian distribution and has a mean of 0.00186 and a standard deviation of 0.102. Because of the normalization of the input data,

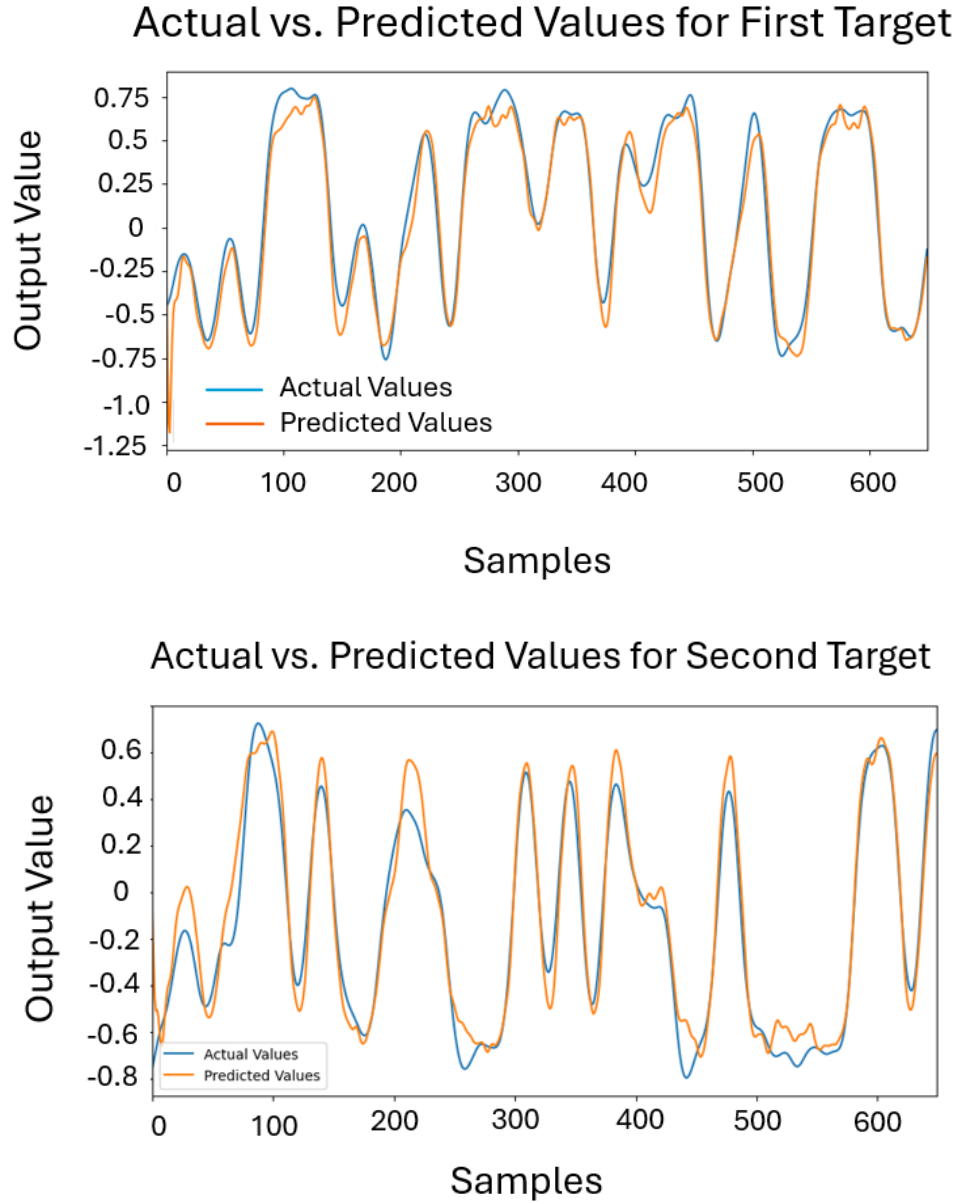


Figure 14: Actual and predicted values for the first target (top) and the second target (bottom). The actual values (T_a and T_b) are obtained from the oscilloscope using the active directional coupler (see Fig. 11). The predicted values are obtained by passively tapping the transmission lines at point A and point B, and feeding the passively obtained information into the trained LSTM. Each trace consisted of 500,000 training points, but the plots show only the first 650 samples.

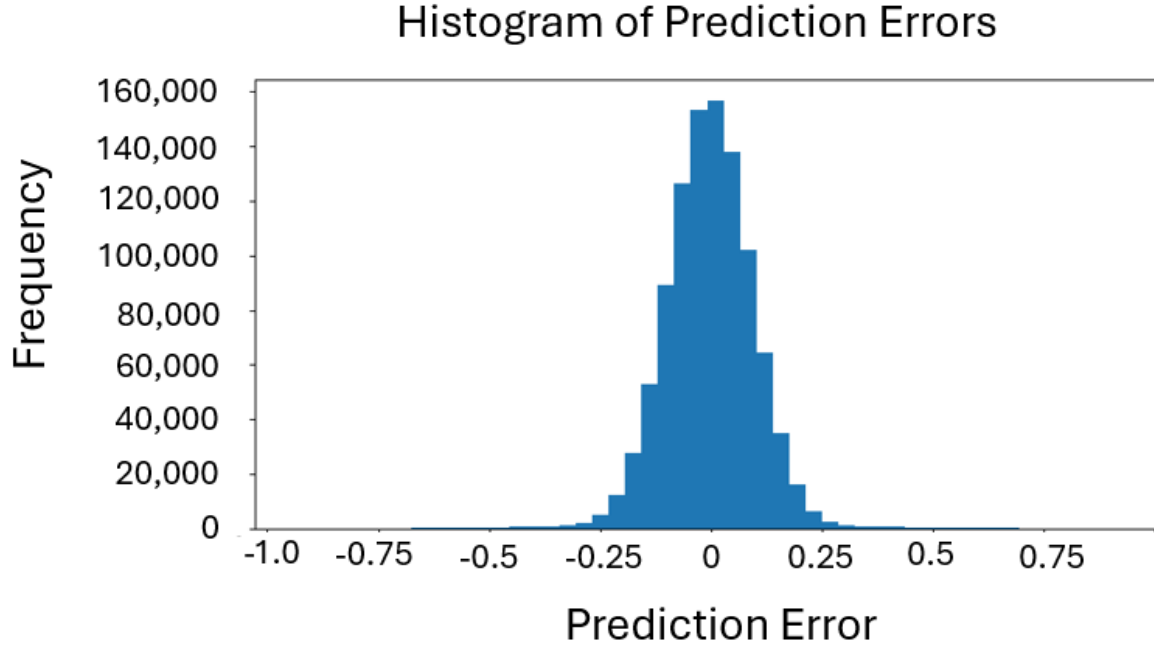


Figure 15: Histogram of point-by-point errors between the actual target values (T_a and T_b) and the predicted target values (derived from feeding A and B into the LSTM). The prediction error has a mean of 0.00186 and a standard deviation of 0.102.

this indicates that about 68% of the samples were correctly predicted within 10% of their actual value, and 95% of samples were correctly predicted within 20% of their actual value.

Conclusion

The machine learning algorithm studied in this research showed success in passive signal separation of the transmit and receive signals in the full-duplex communication. This success indicates future researchers may continue to study the performance of machine learning-style algorithms, specifically LSTM and similar models, on full-duplex communication signal separation. Researchers may discover that machine learning works well on a specific portion of the signal separation, while traditional digital signal processing (DSP) techniques may work well for other aspects. Researchers may be able to take advantage of both approaches.

Because the goal of the signal separation process is to provide signals for decoding, having a voltage/measurement error of 20% may still allow for 100% signal decoding. If the transitions between the three states corresponding to $\{-1\}$, $\{0\}$, and $\{1\}$ are preserved, the decoding procedure may still be successful. One proposed area of future research will be to evaluate the success of decoding the transmission signals after the completed LSTM-based signal separation.

A goal of future research is to develop and train machine learning algorithms capable of passively observing, interpreting, and decoding full-duplex network signals. The research presented here demonstrates the success of an LSTM model in accomplishing this task. Specifically, researchers aim to develop this process to be compatible with Gigabit Ethernet and related communication protocols. The intent is to allow an algorithm to record these signal observations without interrupting or modifying the transmission signals, thus avoiding any disturbance to critical systems.

THESIS CONCLUSION

Research into the potential of machine learning algorithms as tools in science and engineering is a developing field. As modern computing power continues to increase, the potential for applying machine learning technology will also continue to increase. As has been presented in this thesis, there are numerous applications across the scientific disciplines where machine learning driven data analysis may prove beneficial.

From the background information chapter we learn the various disciplines in which machine learning algorithms are being used. From simple data analysis tasks in scientific research to applications in industrial Ethernet communication protocols [6, 8, 14, 18]. Machine learning has led to scientific breakthroughs in physics, astronomy, healthcare, and engineering [13, 25]. Research suggests that machine learning driven data analysis may be viable for most, if not all, technical disciplines. The range of application for machine learning technology is far from strictly defined, and this unknown drives research similar to the work presented in this thesis.

The work studying MODBUS communication in the MODBUS chapter of this paper presents the author's introduction to this style of research. Full-duplex communication is widely used in engineering academia, industry, and throughout the world's communication technology. There is no shortage of applications for passive data acquisition techniques. The success of the research presented in the MODBUS chapter shows promise for future researchers in continuing to study applications of these techniques. The particular experiment presented in the LSTM chapter of this thesis shows promise for potential applications in cybersecurity [33, 36].

When considering a communication of high importance, the relevance of understanding what information may be contained in an incoming transmission seems apparent. However, the data associated with an incoming transmission could contain potentially malicious

information. Passive data acquisition would allow for this knowledge to be obtained without interfering with actual communication. The importance of this technique also seems relevant when considering, for example, time-sensitive communications. It is not beneficial to stop a time-sensitive communication to study the incoming data before fully receiving the essential information of the incoming message.

Initial research using MODBUS communication may lead to similar research using Ethernet communication. A goal of the research presented in this thesis is for researchers to continue studying the LSTM application to Gigabit Ethernet full-duplex communication [33]. The benefit of studying MODBUS communication is a simpler encoding scheme of the transmission data as compared with Ethernet style encoding. This allows for a smoother introduction into this field of research before attempting a project on the more complicated Ethernet style encoding scheme.

The success of the LSTM machine learning model in data acquisition and signal recreation for full-duplex communication may be due, in part, to the time-dependent nature of communication signals. LSTM models have proven effective in time series data analysis and forecasting, based on other research [47]. From the Background chapter we learn of research completed by Banik and colleagues. Their research studies the prediction of electrical energy consumption [6]. Perhaps an LSTM model may be effective in similar research. Machine learning engineers and researchers may determine other similar models to be more effective in the future.

The research presented by Mai and colleagues focuses on determining the application of machine learning algorithms used in scheduling analysis in an effort to create a real-time Ethernet network [32]. Here too, perhaps an LSTM may provide an effective contribution. Mai and colleagues use a KNN algorithm in the presented research. The KNN approach is a simpler method than an LSTM. A KNN algorithm essentially studies the proximity of data points in a particular data set and attempts to make predictions based on these locations.

An LSTM is a deep neural network that can study and accomplish tasks corresponding to more complex aspects of a data set [14, 51].

The research presented by Lucas Torlay focuses on the passive data acquisition from Ethernet devices in the context of cybersecurity of power systems. Torlay employs deep learning to passively authenticate Ethernet devices based on their communication signals. While he reports deep learning proves critical to this method of authentication, in the publicly available abstract he does not specify a particular deep learning model [51]. An LSTM model may be a valid candidate for similar research.

The complexity of full-duplex communication seems a prime candidate for research into machine learning assisted data analysis. The application of the LSTM model for signal separation demonstrates a method that was moderately straightforward to understand by the MSU researchers. An attempt to accomplish this signal separation using a manual decoding method may have proved challenging to K. Filler and others at this time.

The underlying theory of artificial intelligence and machine learning (not to mention the wealth of science fiction material on the subject) can make this topic seem outlandish and daunting. The research experience completed by K. Filler demonstrated how accessible and useful this technology can be. There is no doubt that there are far more advanced aspects to machine learning than presented here, and more complicated problems to solve. This thesis represents an introduction to machine learning and signal processing research for K. Filler. The research proved to be informative and quite tangible after some considerable studying by K. Filler.

Throughout this research, the author has presented several successful experiments using machine learning algorithms for data analysis. The initial MODBUS research presented in the MODBUS chapter of this thesis also reports the shortcomings of a machine learning method compared to a manual decoder method in regard to data analysis of a half-duplex communication. This research suggests that there are many applications where the use of

machine learning may prove beneficial, but there are also applications where traditional DSP techniques are sufficient, or more effective [33].

This is an important observation. It seems that often as new technology emerges, in this case machine learning algorithms, the reaction can be an attempt to apply this new technology to as many areas as possible. The goal of engineering research is not to simply find as many new applications as possible, although that is certainly a goal. Another primary goal is to determine the appropriate boundaries for where the new technology can interact with the time-tested technology.

The application of machine learning to communication protocols seems a viable field for further research. The implementation of a real-time Ethernet network in industry would be a considerable accomplishment [8, 14]. The studies presented in this thesis suggest machine learning driven technology may be critical to this endeavor. The importance of cybersecurity in modern technology is also becoming clear. As our digital technology continues to advance and become more user accessible, modern culture becomes more dependent on digital communication. The security of personal information is an obvious goal moving forward.

The research presented in this thesis demonstrates the success and broad application of deep learning, in particular an LSTM machine learning model [33]. The LSTM model has proven successful in other research using time series data [47]. The success of the LSTM model in analyzing time series data was likely a contributing factor in the initial decision to study this model type with full-duplex communication signals. There may be other deep learning models that perform better in similar future research.

The application for deep learning models to data analysis, compared with standard machine learning models, is a topic of further research. The success of deep learning lies in the ability of these models to process more complex data than, for example, a KNN machine learning model [31]. A simpler machine learning algorithm may be more effective in certain data analysis tasks. These models may be faster trained and accomplish the specific task in

question faster. As the data sets and tasks proposed to machine learning algorithms become more complex, deep learning may become the more effective tool.

Future research involving machine learning applications to communication, and other scientific disciplines, seems quite broad. At this time there seems no particular limit to what researchers can, at least attempt to, apply machine learning algorithms to. As has been previously expressed in this thesis, the relevance of data analysis has increased over the past couple decades as computing power has likewise advanced. Researchers will continue to study these applications and pursue the most efficient combination of traditional and machine learning driven data analysis.

REFERENCES CITED

- [1] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. Internet of things (iot) communication protocols. In *2017 8th International conference on information technology (ICIT)*, pages 685–690. IEEE, 2017.
- [2] FR Ametov, EA Bekirov, and MM Asanov. Organizing the information security in modbus tcp interfaces for use in the energy complex. In *IOP Conference Series: Materials Science and Engineering*, volume 1089, page 012007. IOP Publishing, 2021.
- [3] Muhammad Amjad, Fayaz Akhtar, Mubashir Husain Rehmani, Martin Reisslein, and Tariq Umer. Full-duplex communication in cognitive radio networks: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2158–2191, 2017.
- [4] Simon Duque Anton, Suneetha Kanoor, Daniel Fraunholz, and Hans Dieter Schotten. Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In *Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES '18*, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450364485. doi: 10.1145/3230833.3232818. URL <https://doi.org/10.1145/3230833.3232818>.
- [5] Ercan Avşar and Md Najmul Mowla. Wireless communication protocols in smart agriculture: A review on applications, challenges and future trends. *Ad Hoc Networks*, 136:102982, 2022.
- [6] Rita Banik, Priyanath Das, Srimanta Ray, and Ankur Biswas. Prediction of electrical energy consumption based on machine learning technique. *Springer Nature*, 103, 2021.
- [7] Iñigo Bilbao, Eneko Iradier, Jon Montalbán, Pablo Angueira, Nhan Thanh Nguyen, and Markku Juntti. Deep unfolding-powered analog beamforming for in-band full-duplex. *IEEE Open Journal of the Communications Society*, 2024.
- [8] John Breeden. What is ethernet? a look at its 50-year history and ai-driven roadmap. *Network World*, N/A, 2025.
- [9] Juan Caballero and Dawn Song. Automatic protocol reverse-engineering: Message format extraction and field semantics inference. *Computer Networks*, 57(2):451–474, 2013. ISSN 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2012.08.003>. URL <https://www.sciencedirect.com/science/article/pii/S1389128612002927>. Botnet Activity: Analysis, Detection and Shutdown.
- [10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL <https://doi.org/10.1145/2939672.2939785>.

- [11] Jen-Hao Cheng, Tien-Min Chang, Chung-An Shen, Mohammed E Fouda, and Ahmed M Eltawil. High-throughput independent component analysis processor for full duplex systems. *IEEE Journal on Selected Areas in Communications*, 41(9):2821–2832, 2023.
- [12] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. Achieving single channel, full duplex wireless communication. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom ’10, page 1–12, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450301817. doi: 10.1145/1859995.1859997. URL <https://doi.org/10.1145/1859995.1859997>.
- [13] Colin Jacobs Christopher J. Fluke. Surveying the reach and maturity of machine learning and artificial intelligence in astronomy. *Wiley Interdisciplinary Reviews*, 10, 2019.
- [14] Peter Danielis, Vlado Altmann, Eike Bjoern Schweissguth, Frank Golatowski, and Dirk Timmermann. Survey on real-time communication via ethernet in industrial automation environments. *IEEE Xplore*, n/a, 2014.
- [15] Matheus K Ferst, Hugo FM de Figueiredo, Gustavo Denardin, and Juliano Lopes. Implementation of secure communication with modbus and transport layer security protocols. In *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pages 155–162. IEEE, 2018.
- [16] Keith Filler. "are communication signals time dependant?". *Google AI*, n/a, 2025.
- [17] Mohammed E Fouda, Chung-An Shen, and Ahmed E Eltawil. Blind source separation for full-duplex systems: Potential and challenges. *IEEE Open Journal of the Communications Society*, 2:1379–1389, 2021.
- [18] Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, and Jurgen Schmidhuber. Lstm: A search space odyssey. *IEEE Xplore*, 28, 2016.
- [19] Zhihong Hunter Hong, Liang Zhang, Wei Li, Yiyan Wu, Zhiwen Zhu, Sung-Ik Park, Sungjun Ahn, Sunhyung Kwon, Namho Hur, Eneko Iradier, et al. Blind rf self-interference cancellation for in-band distribution link in atsc 3.0. In *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6. IEEE, 2022.
- [20] Jessica Hopkins. Difference between half-duplex vs. full-duplex. *totalphase.com*, page n/a, 2022.
- [21] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566. IEEE, 2014.

- [22] Torres S. Hugo, Urrea O. Claudio, Kern M. John, and O. Alvarado-Cando. Hamming code to data protection in a modbus industrial communication: Experimental results. In *2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM)*, pages 1–5, 2018. doi: 10.1109/ETCM.2018.8580291.
- [23] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.
- [24] Lucia Keleadile Ketshabetswe, Adamu Murtala Zungeru, Mmoloki Mangwala, Joseph M Chuma, and Boyce Sigweni. Communication protocols for wireless sensor networks: A survey and comparison. *Heliyon*, 5(5), 2019.
- [25] Robert Koch. History of machine learning - a journey through the timeline. *click-worker.com*, 1, 2025.
- [26] Aditya Kumar. What is epoch in machine learning? *Simpli Learn*, n/a, 2024.
- [27] Kevin Lamshöft and Jana Dittmann. Assessment of hidden channel attacks: Targetting modbus/tcp. *IFAC-PapersOnLine*, 53(2):11100–11107, 2020. ISSN 2405-8963. doi: <https://doi.org/10.1016/j.ifacol.2020.12.258>. URL <https://www.sciencedirect.com/science/article/pii/S240589632030536X>. 21st IFAC World Congress.
- [28] Maxwell W Librecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews: Genetics*, 16, 2015.
- [29] Weilin Luo, Ruijuan Yang, Hongbin Jin, Xiaobai Li, Hao Li, and Kangbo Liang. Single channel blind source separation of complex signals based on spatial-temporal fusion deep learning. *IET Radar, Sonar & Navigation*, 17(2):200–211, 2023.
- [30] Xin Lv, Shuyu Wang, Yuliang Zhao, and Peng Shan. A reinforcement learning based method for protein’s differential scanning calorimetry signal separation. *Measurement*, 188:110391, 2022.
- [31] Tieu Long Mai and Nicolas Navet. Improvements to deep-learning-based feasibility prediction of switched ethernet network configurations. *ACM Digital Library*, n/a, 2021.
- [32] Tieu Long Mai, Nicolas Navet, and Jorn Migge. On the use of supervised machine learning for assessing schedulability: application to ethernet tsn. *ACM Digital Library*, n/a, 2019.
- [33] Maximus Marceau, Skyler Reid, and Dr. Bradley M. Whitaker. Passive decoding with machine learning. *MSU Research PowerPoint*, n/a, 2024.
- [34] n/a. Training and validation loss in deep learning. *Baeldung.com*, n/a, 2024.
- [35] Noah Netz and Stephen Ude. Lstm model in jupyter lab. *n/a*, n/a, 2024.

- [36] Stephen Ude Noah Netz. An edge ldrd: Summer intern research. *INL Intern PowerPoint*, n/a, 2024.
- [37] Hideya Ochiai, Md Delwar Hossain, Pawissakan Chirupphapa, Youki Kadobayashi, and Hiroshi Esaki. Modbus/rs-485 attack detection on communication signals with machine learning. *IEEE Communications Magazine*, 61(6):43–49, 2023. doi: 10.1109/MCOM.002.2200553.
- [38] Vaijayanti Panse, Tapan Kumar Jain, Prabhat Kumar Sharma, and Ashwin Kothari. Digital self-interference cancellation in the era of machine learning: A comprehensive review. *Physical Communication*, 50:101526, 2022.
- [39] Brandon Phillips, Eric Gamess, and Sri Krishnaprasad. An evaluation of machine learning-based anomaly detection in a scada system using the modbus protocol. In *Proceedings of the 2020 ACM southeast conference*, pages 188–196, 2020.
- [40] Donovan Porter. 100base-t1 ethernet: the evolution of automotive networking. *Texas Instruments, Techn. Ber*, 2, 2018.
- [41] Panagiotis Radoglou-Grammatikis, Ilias Siniosoglou, Thanasis Liatifis, Anastasios Kourouniadis, Konstantinos Rompolos, and Panagiotis Sarigiannidis. Implementation and detection of modbus cyberattacks. In *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, pages 1–4, 2020. doi: 10.1109/MOCASST49295.2020.9200287.
- [42] Hugo Marcelo Torres Salamea, Dalton Demetrio Toledo Torres, Pablo Dario Urgilés Cardenas, and Claudio Urrea Oñate. Implementation of the hamming code for the detection and correction of errors in a telerobotic system using an industrial communication protocol. In *2020 IEEE ANDESCON*, pages 1–6, 2020. doi: 10.1109/ANDESCON50619.2020.9272049.
- [43] Numukobwa Segolene, Kaiqin Liao, and Wanchun Jiang. Improvement of the prediction-based energy efficient ethernet strategy. *IEEE Xplore*, 7, 2019.
- [44] Chung-An Shen, Mohammed E Fouda, Hsi-Hung Lu, and Ahmed Eltawil. Independent component analysis with nonlinearity mitigation for mimo full-duplex systems. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 993–997. IEEE, 2021.
- [45] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.
- [46] S Vikram Singh, Aizad Khursheed, and Zahoor Alam. Wired communication technologies and networks for smart grid—a review. *Cyber Security in Intelligent Computing and Communications*, pages 183–195, 2022.

- [47] Adrian Tam. Lstm for time series prediction in pytorch. *Machine Learning Mastery*, n/a, 2023.
- [48] Aimin Tang and Xudong Wang. A-duplex: Medium access control for efficient coexistence between full-duplex and half-duplex communications. *IEEE Transactions on wireless communications*, 14(10):5871–5885, 2015.
- [49] George Thomas. Introduction to the modbus protocol. *The Extension*, 9(4):1–4, 2008.
- [50] Lilia Tightiz and Hyosik Yang. A comprehensive review on iot protocols’ features in smart grid communication. *Energies*, 13(11):2762, 2020.
- [51] Lucas Torlay. Analysis of deep learning methods for wired ethernet physical layer security of operational technology. *Clemson Open*, n/a, 2021.
- [52] unknown. Advantages and disadvantages of deep learning. *geeksforgeeks.org*, 1, 2025.
- [53] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM transactions on audio, speech, and language processing*, 26(10):1702–1726, 2018.
- [54] Zhaolin Wang, Xidong Mu, and Yuanwei Liu. Bidirectional integrated sensing and communication: Full-duplex or half-duplex? *IEEE Transactions on Wireless Communications*, 23(8):8184–8199, 2023.
- [55] Lothar Zier, Wolfgang Fischer, and Frank Brockners. Ethernet-based public communication services: challenge and opportunity. *IEEE Xplore*, 42, 2004.