3-23-2018

Cyber Data Anomaly Detection Using Autoencoder Neural Networks

Spencer A. Butt

CYBER DATA ANOMALY DETECTION USING AUTOENCODER NEURAL NETWORKS

THESIS

Spencer A. Butt, Captain, USAF

AFIT-ENS-MS-18-M-113

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States

CYBER DATA ANOMALY DETECTION
USING AUTOENCODER NEURAL NETWORKS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Spencer A. Butt, MS

Captain, USAF

March 2018

AFIT-ENS-MS-18-M-113

CYBER DATA ANOMALY DETECTION
USING AUTOENCODER NEURAL NETWORKS

Spencer A. Butt, BS

Captain, USAF

Committee Membership:

Dr. Bradley Boehmke
Chair

Dr. Kenneth Bauer
Reader

AFIT-ENS-MS-18-M-113

Abstract

The Department of Defense requires a secure presence in the cyber domain to successfully execute its stated mission of deterring war and protecting the security of the United States. With potentially millions of logged network events occurring on defended networks daily, a limited staff of cyber analysts require the capability to identify novel network actions for security adjudication. The detection methodology proposed uses an autoencoder neural network optimized via design of experiments for the identification of anomalous network events. Once trained, each logged network event is analyzed by the neural network and assigned an outlier score. The network events with the largest outlier scores are anomalous and worthy of further review by cyber analysts. This neural network approach can operate in conjunction with alternate tools for outlier detection, enhancing the overall anomaly detection capability of cyber analysts.

For my wife

Acknowledgments

First and foremost I would like to thank my research advisor, Dr. Brad Boehmke, for the time and effort he put in to help me complete this thesis. His considerable contributions greatly improved the quality of this work. I also need to thank Dr. Kenneth Bauer whom first introduced me to neural networks. His early insights and recommendations led me down the research path that would become the analytic foundation for this document. Thanks go to Dr. Raymond Hill for his help with regression analysis.

Spencer A. Butt

Table of Contents

List of Figures

List of Tables

CYBER DATA ANOMALY DETECTION USING AUTOENCODER NEURAL NETWORKS

I. Introduction

The United States military advanced technologies are lucrative targets for exploitation of protected information by adversaries due to the high tactical advantage offered antagonists. Publically acknowledged military cyber targets attacked by Chinese adversaries include, U.S Transportation Command (TRANSCOM) networks, White House computer systems, National Oceanic and Atmospheric Administration (NOAA) weather satellites supporting military operations, Civil Reserve Air Fleet (CRAF) defense contractor computer systems and numerous Department of Defense (DoD) computer systems housing designs for advanced military weapon systems such as the F-35 Joint Strike Fighter, F/A-18, Patriot missile system, Littoral Combat Ship Terminal High Altitude Area Defense (THAAD) missile, and more [1]–[3]. Additionally, civilian infrastructure supporting military readiness are targets for malicious code (“malware”) as evident by the Ukraine power outages in 2015 and 2016 [3], [4].

Recent trends in military adversary tactics have seen a shift from covert exploitations to more overt attacks resulting the widespread dissemination of protected information. In 2016 11.5 million leaked accounting documents from the Mossack Fonseca Panamanian law firm appeared online. In the most recent U.S president election, Democratic Party stolen information was released in an apparent attempt to sway voter opinion [4], [5]. Finally, malicious actors compromised the media company HBO computer networks, exploiting weaknesses in third-party security systems, and managed to exploit 1.5 TB of their data, before announcing the exploitation via email to technology magazine Wired [6].

Independent to the nature of information dissemination gained via cyber espionage, a common theme is found, cyber adversaries extracted protected information prior to cyber-attack detection.

1.1 Motivation

The U.S. heavily relies on the systems of cyberspace and the wider Internet as a whole for commerce, defense operations, infrastructure industrial control systems, financial management, transportation, and other critical services. The DoD is one of many U.S. organizations who bare the shared responsibility for the defense of the U.S. homeland from cyberspace attacks. The DoD currently operates the largest global network, composed piecemeal of thousands of smaller networks stretching across the globe, collectively known as the Department of Defense Information Network (DoDIN). The U.S. military is reliant on a secure and robust DoDIN to conduct is overall mission [3], [7]. The agency sponsoring this research is one of the many organizations responsible for protecting the U.S. from cyber threats and ensuring the DoDIN is available to the DoD when required.

The sponsoring agency currently utilizes firewalls, and various intrusion detection and intrusion prevention systems (IDPS) to detect cyber-attacks. The field of intrusion detection and prevention sole purpose it to detect and defend against malicious network intrusion, with the ultimate goal of preventing hostile actions before malicious code can be executed on defended computer networks [8]. The dynamic and rapidly changing nature of the threats to the sponsor's defended networks necessitated the creation of analysis methods to detect novel and previously undetected attacks. The sponsoring agency's IDPSs generate log files containing details of network traffic identified by the IDPS as suspicious. Creation of suspicious network log files enables data mining intrusion detection techniques for cyber-attack detection. Gutierrez derived techniques for transforming DoDIN IDPS log file data into time-oriented state vectors. His work applied multivariate and graphical analysis techniques for outlier detection [9]. This research will provide security experts a new tool for outlier detection, an autoencoder neural network (ANN).

1.2 Research Goals

Currently, analytic tools available to the sponsor for IDPS cyber-anomaly detection are limited, for example, Gutierrez's derived multivariate and graphical analysis techniques. Additionally, manual review of IDPS log files by computer security experts are also available for cyber log anomaly detection. The number of IDPS log files and the number of new files generated during day-to-day operations of the DoDIN is too massive for manual review by the limited computer security expert staff in any reasonable time frame. It is a widely accepted fact that there does not exist a single best approach for detecting anomalies within data [10]. Commonly in the field of network intrusion prevention multiple analytic approaches are employed to defeat attacks in the evolving threat environment [11]. The sponsor requires a spectrum of analytic options for review and analysis of IDPS log files to maximize the probability that adverse network activity is detected prior to successful adversary misconduct.

This research aims to reduce the time between the start of cyber-attacks by providing the sponsor capabilities to distinguish attacks from normal network activity while minimizing the number of misclassified network events. The use of a cyber anomaly detection ANN will aid the sponsor through expeditious identification of potentially adverse network activity for further evaluation by security experts. Specific research goals include:

1. Derive an ANN to characterize normal network activity and identify anomalous activity in IDPS log file data

2. Develop a methodology to visualize cyber data characterization after ANN analysis

3. Identify specific anomalous IDPS log files for subsequent review by network security experts

There is no guarantee that anomalous IDPS log files are evidence of malicious cyber activity. Identification of anomalous IDPS log files will however provide network cyber security experts a starting point in their search for undetected network intrusions on the DoDIN.

1.3 Research Contributions

The goal of this research is to provide the sponsoring agency an expeditious methodology to identify specific anomalous IDPS observations for review by computer security experts using ANNs. We also provide a simple method for optimizing the performance of the anomaly detection ANN using design of experiments (DOE). Within the field of NN research, we introduce the use of multiple test and training datasets as a hyperparameter combined with DOE testing as a NN regularization technique for unsupervised anomaly detection.

1.4 Assumptions/Limitations

The data used for this research is assumed to be a representative sample of the data the sponsoring agency utilizes for detection of malicious activity during normal day-to-day operations. The dataset is unsupervised, that is without any feature to indicate normal or abnormal observations, therefore we assume that the vast majority of observations within the dataset are observations corresponding to normal network behavior and anomalous observations are rare within the dataset.

1.5 Organization

Chapter 2 provides the reader a literature review related to the field of network intrusion detection, the sponsor's network architecture, intrusion detection terminology, and intrusion detection performance metrics. The literature review continues in the third chapter covering, anomaly definition and types, a survey of anomaly detection methods utilized across multiple domains, neural networks with a focus on hyperparameters, and concludes with a summary of DOE testing methodology. Chapter 4 outlines the methodology utilized in this work to include, dataset description and preprocessing, test design creation, and graphical detection of outliers. Chapter 5 presents the results from ANN optimization and anomaly detection. We conclude in chapter 6 with a brief summary of the contributions to the field of unsupervised anomaly detection in the cyber domain and recommendations for follow-on research.

II. Literature Review: Application Domain

2.1 Chapter Overview

This chapter aims to provide the reader with a brief background into the makeup of the sponsors' defended networks. As these networks are real-world military assets, the network structure presented should be taken as notional, albeit, representative. Following the network description, we introduce Intrusion Detection Systems (IDS) by presenting a model of their operation and briefly discussing required terminology pertaining to their employment. This chapter concludes with a discussion of general IDS output.

2.2 DoDIN Network

There are multiple system components employed to protect computer systems on a network, the firewall, IDS, Intrusion Preventions System (IPS), antivirus software, access control schemes, authentication tools, and virtual private networks [12]–[16]. Despite the surplus of technologies and software available for network security, it is a widely accepted fact amongst network security experts that it is impossible to defend against all possible attacks. Furthermore, it is cost prohibitive and technically infeasible to try and do so [13], [15], [17], [18]. The IDS and IPS provide many of the same capabilities to network administrators, however there is a key difference, an IDS is an automated detector which alerts administrators to the threat in network traffic, whereas an IPS is an IDS with the additional capability to automatically stop potential network security incidents without human intervention [19].

The National Institute of Standards and Technology defines a firewall as "devices or programs that control the flow of network traffic between networks or hosts that employ differing security postures [20]. " Firewalls are software or physical network devices that isolate a network from external sources by preventing traffic external to the network from reaching computer systems on the network. Traditionally, firewalls were deployed only at the boundary between the network and all external sources, and prohibited all inbound traffic not explicitly allowed by the network administrator. Modern firewalls however, monitor and restrict network traffic from exiting (egress filtering) as well as entering (ingress filtering) at multiple points internal to, and at the boundary of, the protected network. Additionally, modern firewalls can strip specific prohibited components from network traffic (i.e. removing email attachments), and perform many of the same functions as IDS and IPS, such as monitoring, threat reaction, and the generation of log files documenting suspicious network traffic for evaluation [12], [20]. Furthermore, firewalls also act as address managers for protected networks, enabling allowed traffic to reach its intended destination within the network, without revealing the detailed internal network structure to external sources [21].

All IDSs perform the same basic function, gather information about the protected system in order to make a determination on the security status and alert network administrators when violations are found automatically in real or near-real time; IPSs, upon detection of a security violation, attempt to take action to mitigate or correct the violation in addition to alert generation. Both IDS and IPS technologies, given a suspicious activity detection, generate log file for subsequent evaluation by network administrators [13], [19], [22]–[25].

The IDS and IPS detection technologies are similar, furthermore, network administrators responsible for IPS implementation may choose to disable automatic prevention features in IPS forcing their operation as IDS. Firewalls also may be configured to generate log files pertaining to potentially adverse network activity. Due to their similarity and to streamline composition, henceforth, we will use the terminology IDPS to refer to IDS, IPS, and appropriately configured firewall systems. Unless specifically stated otherwise, the use of IDPS will refer to any network system or software that "automates the intrusion detection process [19]," and generates an alert (or log) for subsequent analysis by network security experts.

Numerous IDPS sensors are positioned across the DoDIN. When potentially malicious cyber data flows through an IDPS, the sensor generates an alert via a log file. Depending on the vendor, the format of individual IDPS log files may be different between the various IDPS systems on the DoDIN. Utilizing a data formatting tool known as a connector, IDPS log files from individual sensors are reformatted into a common data structure and forwarded to a Regional Cyber Center (RCC) [9]. RCCs maintain, operate, and defend the DoDIN within their respective theater, or area of military operations [26]. RCCs aggregate data from the reporting IDPSs using a Security Information and Event Management (SIEM) system. SIEM systems consolidate IDPS log files, perform threat correlation, identify cyber threats, and report network heath and status information to network security personnel [27]. Once data are aggregated at theater RCCs, the information is passed via additional connectors to the Integration Center's (IC). ICs aggregate data globally from theater RCCs into a global SIEM, and once processed, the global log file data is uploaded into a big data platform. This big data platform is a centralized database for the processing, management, and analysis of the cyber log data [28].

2.3 Intrusion Detection Model

Denning presented the earliest work on detecting unauthorized activity on computer systems using computer methods. Prior to Denning, intrusion-detection consisted of computer security experts manually reviewing audit logs on printouts of computer system files. Printout log files were not examined to prevent intrusions, they were only manually reviewed to close existing vulnerabilities in computer systems [24]. IDS are important security tools that automate intrusion-detection enabling rapid detection of computer compromises and improving the overall security of the protected system [13], [17]–[19], [29]–[31]. The fundamental assumption underpinning all IDS operations is simple, exploitation of a computer system is abnormal and abnormal computer system operation can be detected by computer security analysts [22], [24]. Computer security personnel can use abnormal system activity to detect exploited computer systems. Expedient detection of anomalous activity may lead to mitigation of cyber threats prior to any damage being done to the system. Prior to Denning's seminal work, Anderson derived statistical methods for analyzing computer system audit data. Building on the work of Denning and Anderson, other researchers experimentally showed two important results: analysis of a computer user's pattern of use can be used to discriminate between users, and analysis of computer user behavior can be used to discriminate between normal and anomalous activity [32]–[34].

2.4 IDPS Taxonomy

IDPSs are categorized according to many criteria to include: detection method, audit source, detection paradigm, and monitoring rate [18], [31]. This section provides the reader with an overview of the diverse IDPS technologies. Prior to embarking on the IDPS taxonomy review, it is prudent to discuss the five performance measures used to evaluate IDPS capabilities: accuracy, completeness, performance, timeliness, and fault tolerance [18], [25], [31], [35].

2.4.1 IDPS Performance Metrics

An accurate IDPS correctly categorizes a network attack as such and minimize the number of false alarms, or the number of innocuous detections incorrectly flagged by the IDPS as attacks. Complementary to the concept of accuracy, is that of completeness. A complete IDPS detects all attacks. If an IDPS incorrectly classifies attack traffic as normal the detector has made a false negative declaration [17]–[19], [23], [25], [31], [36]–[39]. Analysis of the IDPS evaluation criteria of accuracy and completeness results in the following conclusion, IDPS declarations can be categorized into one of four possible types, *intrusive but not anomalous*, *not intrusive but anomalous*, *not intrusive and not anomalous*, and *intrusive and anomalous*. Intrusive but not anomalous declarations (false negatives) occur when the IDPS incorrectly categorizes anomalous traffic as normal activity. Not intrusive but anomalous declarations are valid activities incorrectly flagged by the IDPS as suspicious. Not intrusive and not anomalous are declarations by the IDPS which correctly identify valid computer actions as normal activity, and intrusive and anomalous are IDPS declarations which are correctly identified as suspicious computer activity [17].

IDPS performance refers to the rate at which the system can analyze network traffic and dispense classification as legitimate or intrusive traffic. The earliest intrusion detection model required that detection occur in real, or near-real time, therefore, processing speed is critical for successful IDPS implementation [18], [22], [31]. Building on the idea of performance is the concept of timeliness, a timely IDPS will quickly transmit intrusive traffic warnings to the appropriate network security expert to mitigate attacks as quickly as possible. The final IDPS performance measure is fault tolerance. IDPS are a part of the protected network, and therefore should be hardened against cyber-attacks degrading their performance, or to put it another way, tolerant to potential faults caused by cyber intrusion [18], [31].

2.4.2 IDPS Detection Method

The two broad classes of IDPS detectors are *behavior-based*, and *knowledge-based*. Across the literature behavior-based IDPS are also referred to as: anomaly detectors, outlier detectors, novelty detectors, deviation detectors, exception mining, and detection by behavior;

likewise, knowledge-based detectors are also referred to as: signature detectors, misuse-based detectors, attack signature based detectors, and detection by appearance [8], [14]–[18], [25], [30], [31], [36], [38]–[43]. Knowledge-based IDPS use knowledge collected from previously identified network attacks and system vulnerabilities to actively look for network traffic utilizing the previously identified exploits. When a previously categorized network attack is detected, an alarm is raised and the attack mitigated. All network traffic not categorized as an attack is assumed to be acceptable [17], [18], [25], [31], [41]. Alternately, behavior-based IDPS operate on the assumption that network attacks can be identified by observing abnormal behavior of the system. Behavior-based IDPS build a mock model of normal system operation, and any sufficient deviation from the normal pattern of system activity will trigger an alarm. Any novel network behavior, valid or malicious, will trigger behavior-based IDPS to flag the activity as potentially intrusive [17], [18], [25], [31], [41].

Knowledge-based IDPS exhibit excellent accuracy for attack signatures defined with low false alarm rates, however, given a previously unseen attack, a knowledge-based IDPS will fail to classify the new attack as intrusive traffic leading to low completeness. In order for knowledge-based IDPS to detect attacks, system administrators need to ensure that attack definition files are updated regularly, a time consuming task given the rate at which new cyber-attacks are developed globally. However, once a new attack is defined in a knowledge-based IDPS, each new protected system gains the capability to defend networks against the identified attack immediately upon instillation. Knowledge-based IDPS detected attacks provide network security personnel detailed contextual analysis associated with the attack traffic, enabling security personnel to easily make preventive or corrective actions on the network. Additionally, knowledge-based IDPS have excellent fault-tolerance to previously defined attack traffic, however, undefined and novel cyber-attacks have the capability to degrade knowledge-based IDPS performance [8], [17], [18], [25], [31].

Behavior-based IDPS offer network security enhanced completeness through the capability to detect previously unknown attacks, and detection of intrusions occurring within the protected network (insider threats). Additionally, behavior-based IDPS show improved fault-tolerance, as they are trained on normal network activity and malicious actors do not know what network activities will trigger an alarm. However, as behavior-based IDPS require training on normal activity, there exists the possibility that nefarious network activity is present in the data used to build the model of normal activity. As a result the IDPS will learn malicious activity as normal, decreasing the behavior-based accuracy. Furthermore, network activity is constantly changing and evolving, requiring behavior-based IDPS retraining on new activity, a time consuming process. Behavior-based detectors are prone to lower accuracy, exhibited by large numbers of false positive declarations. Depending on the number and frequency of false positive, it is possible that network security experts will ignore a true attack alarm, attributing it to another false positive. Identification of the specific network traffic causing alarms in behavior-based IDPS require review by network security personnel prior to classification as either approved but anomalous network activity or, malicious activity [8], [17], [18], [25], [31]. Behavior-based IDPS can contribute to the automatic detection of new attacks, although network security experts must manually review suspect network traffic to make the final classification [18].

2.4.3 IDPS Audit Source

IDPS audit sources describe the source of the information, known as an audit log, utilized by the system for security dispensation. IDPS are broadly divided into *host-based* and *network-based* security systems.

Host-based IDPS monitor and analyze data collected on individual computers, monitoring the instructions moving through the computer's processor, logins, program execution, file access history, etc. As their name implies, host-based IDPS are installed on individual computer systems they are designed to protect. Host-based IDPS require excellent completeness, timeliness and fault tolerance, and generally are individually tailored to defend a specific system. If a computer protected with a host-based IDPS is compromised by a malicious actor, and the compromise is not expediently detected, the compromising party may alter the computer log files and/or modify the IDPS to prevent any/all detection of subsequent malicious activity. As the earliest computer systems moved from standalone into a distributed architecture, the IDPS architecture followed suit moving from host-based IDPS software installed on individual machines, to independent hardware/software within the network environment [8], [18], [31], [33], [40].

Network-based IDPS monitor and analyze data collected moving between computer systems connected on a network. In the modern computer environment, the majority of access to sensitive computer systems takes place over a computer network, necessitating the creation of specialized systems to monitor traffic and identify hostile traffic [31]. Network-based IDPS utilize the data contained within network packets to form the audit log. Analysis of the network packets reveal malicious traffic on the network. Network-based IDPS are standalone units that commonly operate without communicating with other sensors on the network, and are often not the target of malicious traffic. As a result, the true target of the attack can often be obfuscated leading to decreases in accuracy and completeness. Additionally, as the size of networks continue to grow and traffic on networks rise, the use of network-based IDPS may induce bottlenecks resulting in decreases in performance and timeliness [8], [18], [31], [40].

2.4.4 IDPS Monitoring Rate

In addition to the audit source and detection method IDPS classifiers, the monitoring rate is used to categorize IDPSs. The monitoring rate refers to the way the system performs analysis on the audit source. A continuous or dynamic monitoring IDPS operates in real-time, processing each individual audit source event immediately upon receipt. The continuous monitoring system continuously analyzes events in real-time and determines the hostility of each individual event immediately. Static or period IDPSs do not operate in real-time, instead, they aggregate a batch of audit source records and on a periodic basis evaluate the grouping of records for security dispensation [18], [31].

2.5 IDPS on Distributed Networks

As previously discussed, the sponsor operates a series of IDPSs located across their area of responsibility. Regardless of the specific taxonomy of each individual IDPS, suspect activities are logged and data is aggregated via connectors to RCCs, the IC, and then the log files are uploaded to the big data platform for analysis. Traditional network intrusion detection uses independently operated IDPSs to defend a single computer or a standalone network. The implementation of networked computer systems connected to the wider internet has led to the development of cooperative attacks against individual systems or networks of systems.

A cooperative intrusion attack differs from a conventional attack in that a conventional attack has one malicious actor targeting a computer or single networked system via one attack vector, whereas in a cooperative attack there are multiple attack vectors. One attacker may use a series of machines located around the world, or multiple attackers may coordinate efforts to exploit a computer or networked system of machines, spreading attacks across multiple IDPS sensors. The goal of the cooperative attack is to defeat conventional IDPS by diffusing the attack across multiple IDPS sensors to obfuscate detection of the full attack taking advantage of the fact that often IDPSs operate independently, not aware of activities occurring on other IDPS sensors within the defended system [13], [41].

Detecting cooperative attacks across a distributed network system requires collecting suspect activities from each IDPS operating within the defended system. Once the data is collected, the log files can be aggregated, correlated and analyzed from each IDPS. Two primary methods exist for analyzing IDPS data collected across the distributed network system, *centralized* analysis and *decentralized* or *hierarchical* analysis. In centralized network intrusion detection, all of the data from each IDPS sensor is collected to a single location for intrusion detection analysis. In hierarchical network intrusion detection log files from each IDPS is first aggregated into a domain, then forwarded to a single location for analysis. A domain is a grouping of IDPS sensors located within the full decentralized network. IDPS sensors can be grouped by geographic area, administrative control, commonality of protected systems and/or required security for information on the network. Once data aggregation is completed at the domain level, analysis is completed on the IDPS sensor information, and this information is then forwarded up to a single location for follow-on analysis [41].

The sponsor operates many types and versions of IDPS sensors across their distributed networks. The collection of IDPS sensors are from different vendors, running differing software versions, record security events in differing formats, etc. and as a result there does not exist a consistent message format between IDPSs at the device level. The sponsor utilizes the *ArcSight Common Event Format (CEF)* to support aggregation of security log files from the numerous IDPS sensors into the big data platform. The CEF provides the sponsoring agency a standardized open log management format to consolidate applicable security logs into a common format for analysis by network security experts [44]. Once security logs are collected into the centralized big data database, the sponsor security experts can query the database for security investigation.

2.6 IDPS Analysis

The sponsoring agency consolidates IDPS security log files into the big data platform daily. Each day, thousands to millions of security events take place on the sponsor's distributed networks and each of these events generates an IDPS log file. IDPS log files describe network traffic which has been already classified as potentially hazardous, therefore the big data platform is composed only of log files describing potentially malicious network actions. How then should analysts parse down the daily thousands-to-millions of IDPS log files into a handful of logs for review by network security experts?

Denning's intrusion-detection model is predicated on the assumption that computer security incidents are abnormal events when compared to the normal profile of computer usage [33]. If we consider the spectrum of security events logged in the big data platform in a similar manner to Denning's perspective, by detecting those abnormal security events within the IDPS log database we will detect those network security worthy of attention by network security analysts. This view is predicated on the assumption that the most common network security events are less threatening than rare network security events. While validating this assumption is beyond the scope of this work, Chandola et al. [36] support this claim and state "anomalies in data translate to significant (and often critical) actionable information in a wide variety of application domains. For example, an anomalous traffic pattern in a computer network could mean that a hacked computer is sending out sensitive data to an unauthorized destination." Also supported by Hodge and Austin [45] who state "An outlier may pinpoint an intruder inside a system with malicious intentions so rapid detection is essential." In order to expeditiously identify potentially harmful attacks against the sponsor’s networks, analysts must identify anomalous observations within the IDPS log files daily.

III. Literature Review: Anomaly Detection Methods

3.1 Chapter Overview

This chapter summarizes common anomaly detection methodologies focusing on those methods utilized in the field of cyber intrusion detection. After summarizing the various methodologies, we focus on neural network (NN) approaches to anomaly detection. We conclude this chapter with a brief summary of DOE and its application for identification of the optimum hyperparameters for anomaly detection using NNs. To begin we first define anomaly detection, identify the characteristics that make observations anomalies, and discuss supervised and unsupervised datasets.

3.2 Anomaly Definition

Anomaly detection refers to finding specific data points, or observations, located within datasets with characteristics not present elsewhere in the data. Specific non-characteristic observations are called anomalies, outliers, novel events, noise, exceptions, etc. depending on the specific application domain [10], [36], [45]. Within this text we will refer to anomalies and outliers interchangeably and we utilize the outlier definition established by Grubbs [46]: "An outlying observation, or "outlier," is one that appears to deviate markedly from other members of the sample in which it occurs."

Anomalies within dataset can be attributed to one of two possible sources. An outlying point may be an extreme, albeit normal, value of the system under study. Given a dataset is large enough, analysts expect to see extreme valued observations as most systems worthy of study exhibit variability in measures of interest to researchers. For example, a far-northern city experiencing a day of unusual warmth during the winter months. While this may be a rare occurrence, given a long enough period of study, we would expect to find periods with temperatures sufficiently different than the mean as to indicate an outlier. Alternately, an outlier may be attributed to some systemic failure of the experimental process or transcription error. For example, a temperature sensor failing in a vehicle engine, or a transcription error while recording data. We consider intrusion detection, or attempted intrusions, as a systemic failure of the system. While anomalous observations must be attributable to one of the previously discussed sources, domain expertise is required to determine to which source the anomaly should be assigned [46].

Observations identified as outliers exhibit two properties. First, anomalous observations are rare occurrences within the data. This property is somewhat obvious, given a sufficient number of the same type of anomalous observations exist within a dataset, when taken collectively these ‘anomalies’ would form a normal pattern within the data, and thus not be anomalies. Second, anomalous points are distinct and sufficiently different from other observations in the data with respect to the features used to describe the collection of observations [36], [47]. If a potential anomalous point, described by features in the data is indistinguishable from the other points in the data, then this point is merely a normal point.

There are two primary anomaly taxonomies, *local* and *global.* Local anomalies can be subdivided into *contextual (conditional),* and *collective* anomalies. Global anomalies can be found by observing the entirety of the available dataset, whereas local anomalies require evaluation of points in relation to neighboring points [47]. The classification of anomalies into their appropriate taxonomy is data dependent. Global (point) anomalies can occur in any dataset, however local anomalies require that data contained with the set be somewhat related. Point anomalies can easily be considered contextual anomalies if the points are analyzed with respect to additional features within the data. Point anomalies can also be considered collective anomalies if considered within a collection of related data [36].

An anomaly may be characterized as a global (also known as *point*) anomaly if the individual observation is sufficiently different than the rest of the data. Global anomalies are the simplest and the vast majority of research on anomaly detection is devoted to identification of point anomalies [36], [47]. An example of a global anomaly is shown in Figure 1.

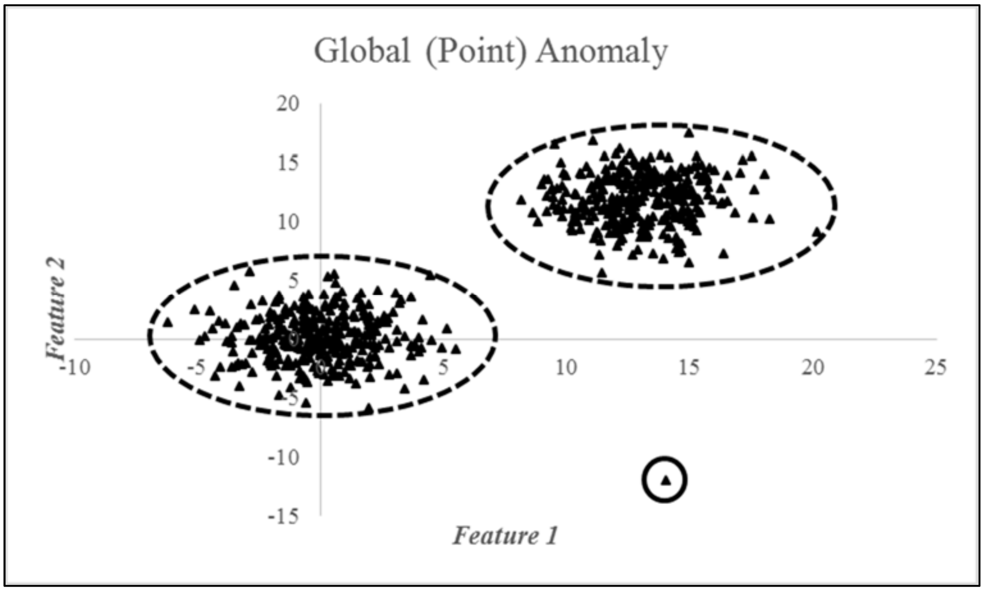


Figure 1. Global (Point) Anomaly

The data points contained with the two dashed ellipses indicate the existence of two discrete clusters of observations. In the Figure 1 example, the data points are described by *Feature 1* and *Feature 2*. The data point encircled by the solid black line is an example of a global (point) anomaly, as its location in the feature-space is far separated from the two clusters, and, there exists only one observation in the local neighborhood.

A contextual anomaly differs from a point anomaly in that a contextual anomaly is only anomalous in some contexts, but not in others. The contextual component of the anomaly is the structure of the data in the neighborhood of the anomalous point, what makes the point an outlier is the behavior of that point in relation to that of its local context [36], [47]. Figure 2 depicts a notional plot of *Feature 1* and *Feature 2*.

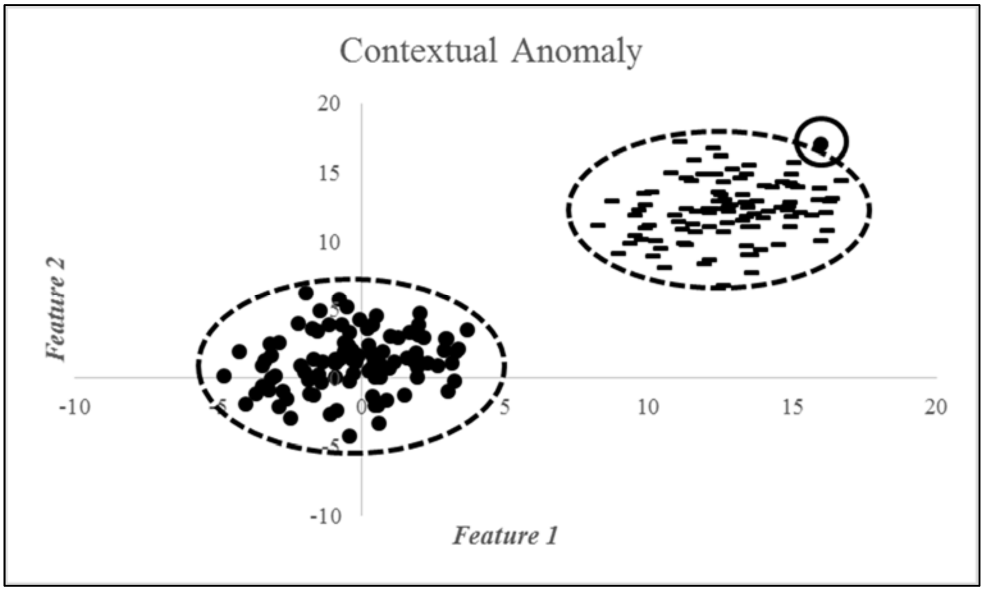


Figure 2. Contextual Anomaly

Observe the groupings of like-points contained within the dashed circles. Note that a circle point is located adjacent to the dash point grouping, highlighted by a solid circle, this circular-point is a contextual anomaly due to its location adjacent to the dashed-point grouping. The anomalous point is only anomalous *conditioned* on the type of neighboring points. If the anomalous point was located within the circular-points region, this point would no longer be anomalous. Also consider the figure if the dashed-points were removed from the plot as shown in Figure 3.

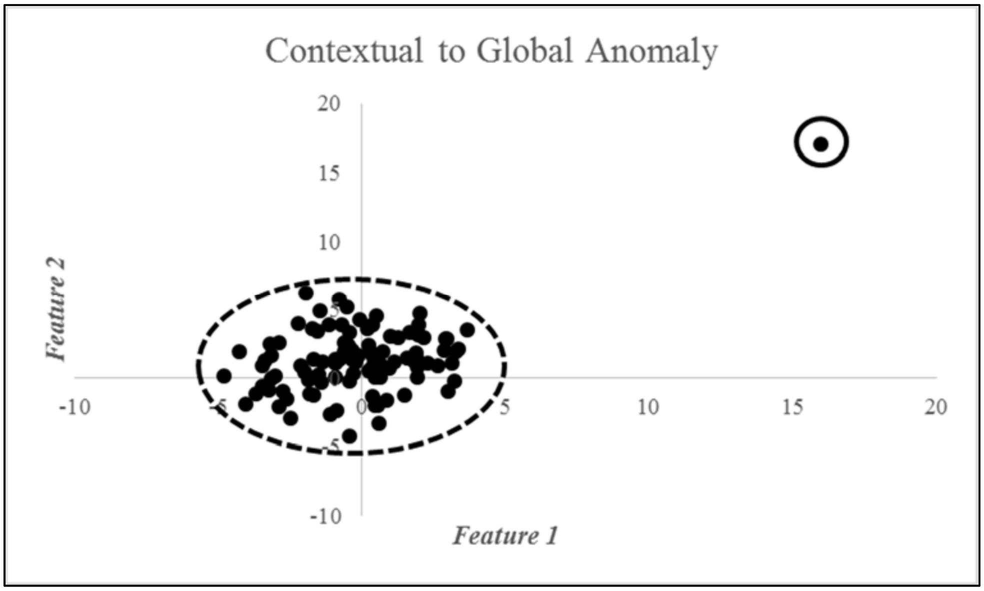


Figure 3. Contextual to Global Anomaly

In this case, the contextual anomalous point would still be considered an anomaly, however, now we would categorize this point as a global (point) anomaly as the context of the removed dashed-points is missing.

A collective anomaly is a set of data points that is anomalous in relation to the entirety of the dataset. Each individual point within a collective anomaly may be normal in relation to the dataset, however the proximity of these points to one another in relation to the behavior of the data across the set results in a collective anomaly [36], [47]. Figure 4 depicts a notional plot of *temperature* as a function of *time*.

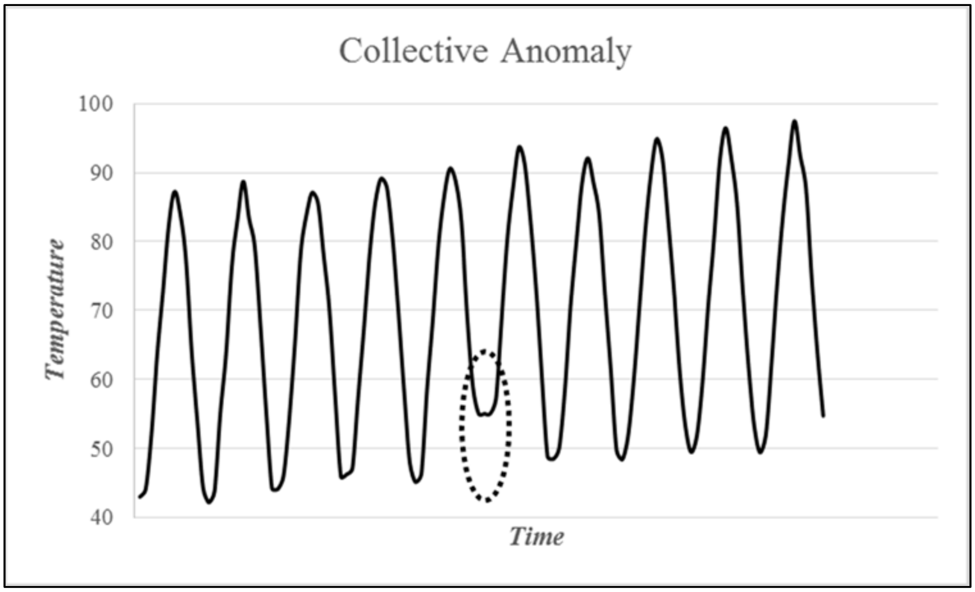


Figure 4. Collective Anomaly

The area encircled by the dashed-ellipse is an example of a collective anomaly. Observe the periodic nature of the *temperature* across the *time* axis, there are regular occurring local maximums and minimums. This normal pattern is not observed for the cycle identified within the dashed-ellipse. This is an example of a collective anomaly. Please note that the temperatures in this anomalous region fall with the normal range across the entirety of the dataset, what makes these points anomalous is their collective contribution in distorting the normal periodic nature of the *temperature-time* plot. Figure 5 depicts an example of a global, contextual and collective anomaly.

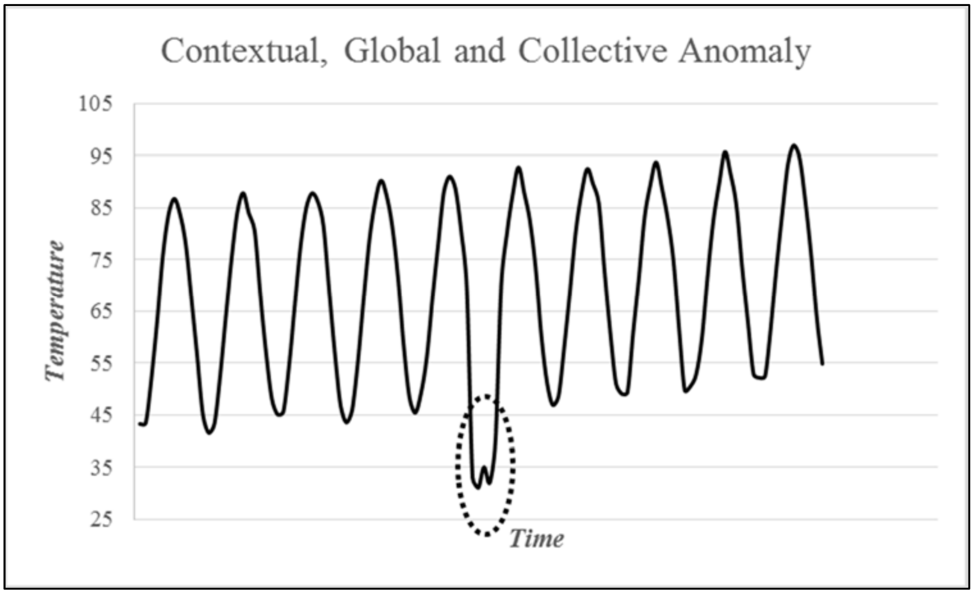


Figure 5. Collective, Global, and Global Anomaly

Observe the anomalous region of the figure highlighted by the dashed-ellipse. Now we have *temperature* values that fall greatly ouside the range of the *temperature* measures elsewhere in in the data indicating a global (point) anomaly. Addiditonally, we observe two local minimums seperated by a local maximum in rapid succession, with all other local minimums occuring in regular periodic intervals. Figure 5 depicts a notional case where we are able to justify the anomalous points as being of all three anomaly taxonomies.

The selection of an appropriate detection methodology is dependent on the type of anomalies expected to be found in the data [36]. Ideally, we would like to apply an anomaly detection methodology with the capability to detect all classes of anomalies (point, contextual, and collective) with equal capability. In practice however, the capability of anomaly detection methods to find outliers is dependent on the class of anomalies present in the data. Analysts must carefully consider the nature of the expected anomalies prior to selection of an anomaly detection mythology. Another important consideration to the choice of an anomaly detection method is the availability and type of data.

3.3 Supervised and Unsupervised Data

Anomaly detection algorithms are often stratified into one of three methodologies based on the type and availability of labeled data contained within the set. Labeling data in the context of anomaly detection refers to identifying those observations within the dataset that are normal and those that are outliers, also referred to as classifying data as normal or anomalous. This action is often performed by a human expert with experience in the application domain pertaining to the dataset. Depending on the domain, it may be impossible to obtain labeled data for a representative set of all possible observations for both normal and anomalous cases, or it may be prohibitively expensive to do so. Additionally, anomalous observations often vary over a period of time [36]. There are many domains in which labeled data does not exist prior to anomaly detection. The three anomaly detection methodologies are *supervised anomaly detection*, *semi-supervised anomaly detection*, and *unsupervised anomaly detection*.

Supervised anomaly detection methods assume that there exists a fully labeled representative dataset available. Generally, the anomaly detection algorithm is trained against the labeled dataset to build a predictive model. Then a second dataset of unlabeled data is evaluated using the predictive model to identify normal and anomalous observations [36]. Often the labeled data is referred to as *training data*, and the unlabeled data as *test data* [47]. Authors caution, supervised anomaly detection algorithms require static data. If the distribution of test data shifts (often over time), then the supervised anomaly detection model must retrain on a new representative training (labeled) dataset, which may or may not be available [45].

The availability of labeled data for all possible anomalous observations generally do not exist. Often datasets only contain observations from the normal class are available. In this case semi-supervised anomaly detection algorithms are used to detect outliers. Generally, semi-supervised anomaly detection algorithms are trained on normal-only datasets to construct a predictive model. Then, test data is presented, and observations deviating from the semi-supervised predictive model are marked as anomalous observations [36], [47]. Semi-supervised anomaly detection (also called one-class classification) defines a boundary of normal observations, and any observation falling outside the boundary of normality is defined as an outlier. Like the supervised anomaly detection methods, if the normal boundary is not static, then the training algorithm must re-learn a new model on representative data [45].

The final anomaly detection methodology is the unsupervised case, where labeled data does not exist. There is no training set, only the test set is available and within the test dataset instances of both normal and anomalous observations may be present. Unsupervised anomaly detection algorithms are required to differentiate normal and abnormal observations using the fundamental attributes of the analyzed dataset. Generally, the unsupervised anomaly detection methods assume there exist far fewer anomalous observations than normal observations. If this assumption is violated, it is common to see an overly sensitive algorithm resulting in a high number of normal points categorized as anomalous [36], [47]. Unsupervised anomaly detection algorithms are forced to define a normal pattern. Observations falling within the normal pattern are marked as normal, and observations falling outside the normal pattern as anomalous [45].

3.4 Methods

Anomaly detection is a challenging problem widely studied across multiple domains of research. Anomaly detection methodologies generally proceeded in two phases. In the first phase, we first define what constitutes normal behavioral patterns for the observations in the data [36], [47]. The identification of the patterns within the data is commonly referred to as *pattern recognition* [11]. In the second phase those points (anomalies) that do not match the normal observations are identified as outliers. Anomaly detection methods generally output scores or labels for each of the observations. Labeling-methods assign a qualitative (categorical) identifier to each observation, generally normal-observation or anomalous-observation. Score-based methods assign a quantitative value to each of the observations. Then domain-specific expertise is applied to set a decision value, where scores less than the decision value are marked as normal-observations, and values greater than the decision value are marked as anomalous-observations [36], [47].

Prior to selecting the most appropriate anomaly detection method, analysts must take multiple factors into consideration. First, they must consider the type of anomaly likely to be present in the data, global, collective, and/or contextual. As discussed previously, there does not exist a singular-best anomaly detection mythology for all anomaly-types. Secondly, the type and amount of data available to the analysts is reviewed. Is the data supervised, semi-supervised or unsupervised, additionally, is the data numeric, categorical, or mixed between the two? Often analysts first examine previous anomaly detection methods employed in the domain specific area of research.

Chandola et al. [36] provide a survey of numerous anomaly detection methods across multiple domains to include methods used in network intrusion detection. Ahmed et al. [43] build on Chandola et al.'s work, providing a more detailed study of the anomaly detection methods for network intrusion. Patcha and Park [17] present the anomaly detection methods of numerous commercial available IDPSs available to network administrators. Goldstein and

Uchida [47], present a detailed review of unsupervised anomaly detection methods across a variety of application domains. Hodge and Austin, Agyemang et al., Markou and Singh, and Beckman and Cook [10], [45], [48]–[50] likewise provide a summary of multiple anomaly detection methods for supervised, unsupervised, and semi-supervised data. Within these resources, the various strengths and weakness of the discussed anomaly detection methods are compared and contrasted to one another.

Across the anomaly detection literature NN anomaly detectors are widely viewed to have benefits above other detection methods. NNs (also commonly referred to as artificial neural networks) have shown the capability to learn the complex relationship differentiating normal from anomalous observations, using linear and non-linear combinations of the data features [11], [16], [45]. Furthermore, NNs are widely adaptable to multiple domains, requiring little domain expertise for implementation [11]. Generally, NNs make weak assumptions regarding the distributions of the features [10], [51]. Alternate anomaly detection algorithms, such as statistical-based approaches, require strong assumptions regarding the underlying distribution of the features, limiting their applicability in the diverse set of features present in network anomaly detection. Conversely, NNs adapt to the provided data structure and learn the potentially complex relationships between features used to differentiate outliers from the normal data [11]. NN approaches have shown much promise in unsupervised anomaly detection. Whereas conventional anomaly detection attempts to differentiate between two (or more) classes, NN approaches develop conceptual representations of the data. Then, NN methods learn to differentiate between the learned representations [52]. Due to their capability, NNs have seen wide adoption as anomaly detectors in a variety of domain areas, Table 1 summarizes specific instances of their usage.

Table 1. NN Anomaly Detection Domain Areas, summarized from [34], [44], [49]

Anomaly detection area	**Reference**
Host-Based Intrusion	[16]
Network-Based Intrusion	[15], [53]–[60]
Credit Card Fraud	[61]–[63]
Mobile Phone Fraud	[64], [65]
Mechanical Fault	[60], [66]–[73]
Structural Damage	[74]–[76]
Image Processing	[60], [77]–[82]
Topics in Text Data	[83]
Medical and Public Health	[60], [66], [69], [84]
Robotics	[85]
Manufacturing	[86]
Object Recognition in Video	[79]

Within the field of network anomaly detection, NNs are widely utilized as evident from Table 1. Chandola et al. [36], cites 47 different anomaly detection applications of NNs, seven (of the 47) of which are in the domain of network anomaly detection, and 14 (of the 47) which are implemented using ANN architecture. The prolific use of ANNs in multiple domains indicate their superior capability as anomaly detectors. Markou and Singh [50] remark "The most striking ability of the auto-encoder is the ability to implicitly learn the underlying characteristics of the input data without any *a priori* knowledge or assumptions." The authors summarize ANN uses for anomaly detection in a variety of domains to include: detecting shorted windings in operational turbine-generators, damage detection in offshore platforms, structural damage detection in bridges, crack detection in structural beams, document classification, and detecting changing environmental conditions. Japkowicz et al. [52] compared more traditional NN anomaly detection approaches to the ANN. Their work examined CH-46 helicopter gearbox faults, biologic molecular promotor recognition, and target identification in sonar data. The ANN approaches outperformed traditional NN approaches on the helicopter and sonar target domains, and performed comparably in the molecular promotor detection task. Within the network intrusion detection domain, numerous authors have document successful ANN implementation for anomaly detection for the purpose of detecting hostile network traffic to include: Hawkins et al. [56], Williams et al. [87], Dondo et al. [88], [89], Dondo and Treurniet [90], Mukhopadhyay et al. [91], Tuor et al. [92], and Mukhopadhyay and Chakraborty [93]. The availability of numerous sources concerning the application of ANN within the field of network anomaly detection is a strong indicator to their efficacy. ANN application in the numerous diverse aforementioned domain areas underlies their capability regardless of the characteristics of the underlying datasets.

3.5 Artificial Neural Networks

3.5.1 Background

NN's are mathematical algorithms designed to complete a task via similar methods as present in the mammalian brain. As opposed to being explicitly designed to solve a particular problem, NNs (and artificial intelligence algorithms in general), gradually improve their own performance on a given analytic task. NNs acquire knowledge by extracting useful information from the data they are provided, through a process called learning (or training). As they learn, NNs adjust their own parameters to improve their performance [94]. Not being explicitly programmed for a specific task, NNs are widely adaptable to a variety of problems in many domain areas.

NNs were originally intended to be a mathematical model of the way in which the mammalian brain solves tasks and learns. The purpose of using the mammalian brain as inspiration for an analytic computational task is twofold. First, the animal brain is intelligent and capable of solving a variety of complex tasks and problems using a diverse set of data. In designing an algorithm which mimics the animal brain, early researchers hoped to recreate the functional capabilities of the animal brain in an artificial (computer) environment. Second, neuroscience researchers are deeply motivated to understand the brain, specifically, how a dense cluster of interconnected unintelligent cells (neurons) are capable of working together to produce intelligence. Neuroscientific researchers hoped by understanding how NNs learn to solve problems and complete tasks, they would also obtain a greater understanding of the process by which biologic learning occurs in animal brains [95].

Modern neuroscientific researchers no longer view NNs as an accurate method to describe how the animal brain learns. Currently, there is not sufficient evidence to support the claim that NN accurately model animal cognition. Even in some of the most well studied areas of the brain, such as the area responsible for vision, there exist outstanding questions regarding how the brain operates and learns [96]. The field of modern cognitive neuroscience is currently researching the algorithms which govern learning in the animal brain. Recent research [97] in the field indicates that a single, yet unknown, algorithm is responsible for solving all the problems and tasks for which the animal brain is responsible. Additionally, neuroscientific research indicates that animal intelligence is a function of the interactions between the individual biologic-computational units of the brain (neurons) with one another [95]. The two important insights gained from neuroscientific research, the single algorithm responsible for learning globally across the animal brain, and intelligence through interaction between unintelligent individual units, are the primary biologic influences for modern NN implementation. For both NNs and biologic intelligence, a large number of simple information processing units learn, and once trained, are collectively able to solve complex problems and tasks [98].

3.5.2 Biologic Neuron

The basic unintelligent information processing unit of the mammalian brain is the biologic neuron, shown in Figure 6.

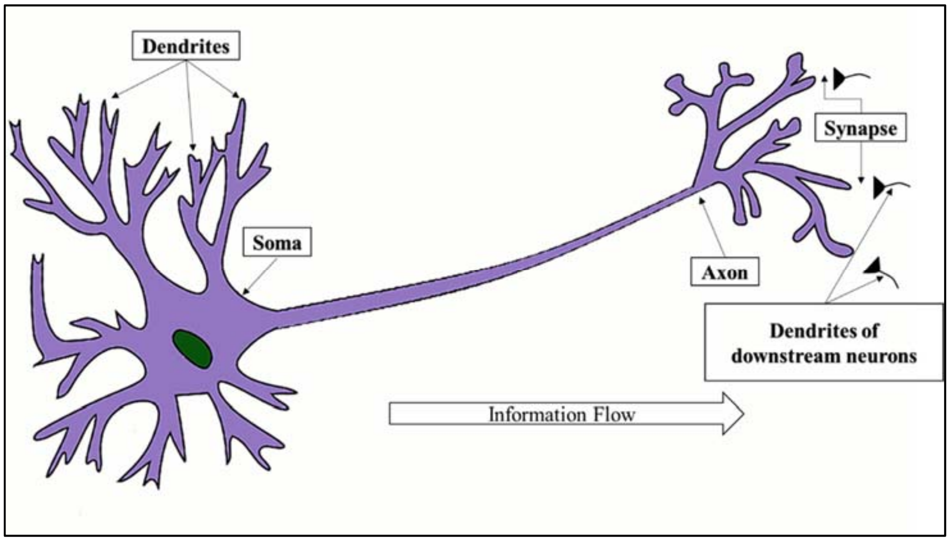


Figure 6. Biologic Neuron adapted from [99]

The neuron is composed of three primary components, the soma, dendrites, and axon. The soma is the primary structural unit of the neuron, containing the cell's nucleus (in green), and the material required for transmitting biological signals. The neuron's dendrites and axons connect neurons to one another. Dendrites receive signals from preceding neurons upstream and axons transmit information via electrochemical signals to subsequent neurons down streams. Axons connect to dendrites through synapses, which are gaps in which electrochemical signals flow from axons to dendrites. Biologic learning is accomplished via adjusting the effectiveness of the signal transmission within individual synapses. Within the mammalian brain it is common to see neurons interconnected to hundreds or even thousands of other neurons [94].

Early cognitive neuroscientific research into the operation of the biologic neuron assessed that each neuron exhibited three characteristics which early NN practitioners attempted to replicate. First, biologic neurons accumulate input signals from each connected neuron upstream. Second, biologic neurons operate as all-or-nothing units. A neuron will only generates a response signal if a sufficient number of input signals are received. Third, biologic neurons are capable of adjusting the efficiency of individual synapses. While subsequent research indicates the operation of individual biologic neurons to be more complex than originally assessed, these original operating characteristics are present in current implementations of NNs [100].

3.5.3 Artificial Neuron

The artificial neuron, the analog to the biologic neuron, is shown in in Figure 7. Information represented as the input signals x_j $(j = 1, 2, \dots, p)$, flows into the neuron from an outside source. Each x_j may be an individual observation within a dataset, however the input signals into an artificial neuron may also be the output of a previous neuron upstream. We use $\vec{x}_i$ to represent an individual observation of p-features from the full dataset $\boldsymbol{x} \in \mathbb{R}^{n \times p}$ for n data observations. Once a data observation ($\vec{x}_i \in \mathbb{R}^{1 \times p}$) enters the artificial neuron, each feature within the data observation is scaled by an individual weight. The individual weights on the input signals function as the biologic neurons synaptic efficiency adjustment. Once the input signals are scaled, the total contribution of all scaled input signals are summed. The summation accumulates the contribution of each individually adjusted input signal as in the biologic neuron. Once summed, the cumulative weights are then evaluated using an activation function, φ. The activation function is the method by which the all-or-nothing behavior of the biologic neuron is implemented in artificial neurons.

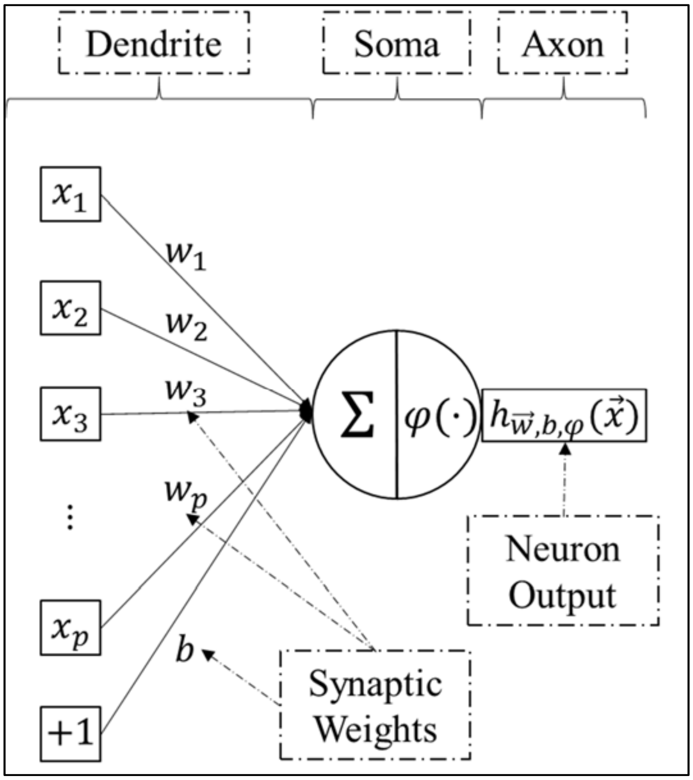


Figure 7. Artificial Neuron adapted from [101]

The output from the artificial neuron, denoted $h_{\vec{w},b,\varphi}(\vec{x}_i)$, is a function of the data observation and the weights, bias and choice of activation function (φ). Note the existence of the b weight associated with the +1 term known as the bias. This term can be thought of as the intercept of the data, analogous to the intercept-term associated with linear regression [102].

3.5.4 Feedforward Neural Network

An individual artificial neuron has little capability to solve a complex problem, however networks of artificial neurons have shown remarkable capability to solve a variety of problems. The classic NN model is the called the feedforward neural network (FNN) (also called multilayer perceptron (MLP)) and is shown in Figure 8.

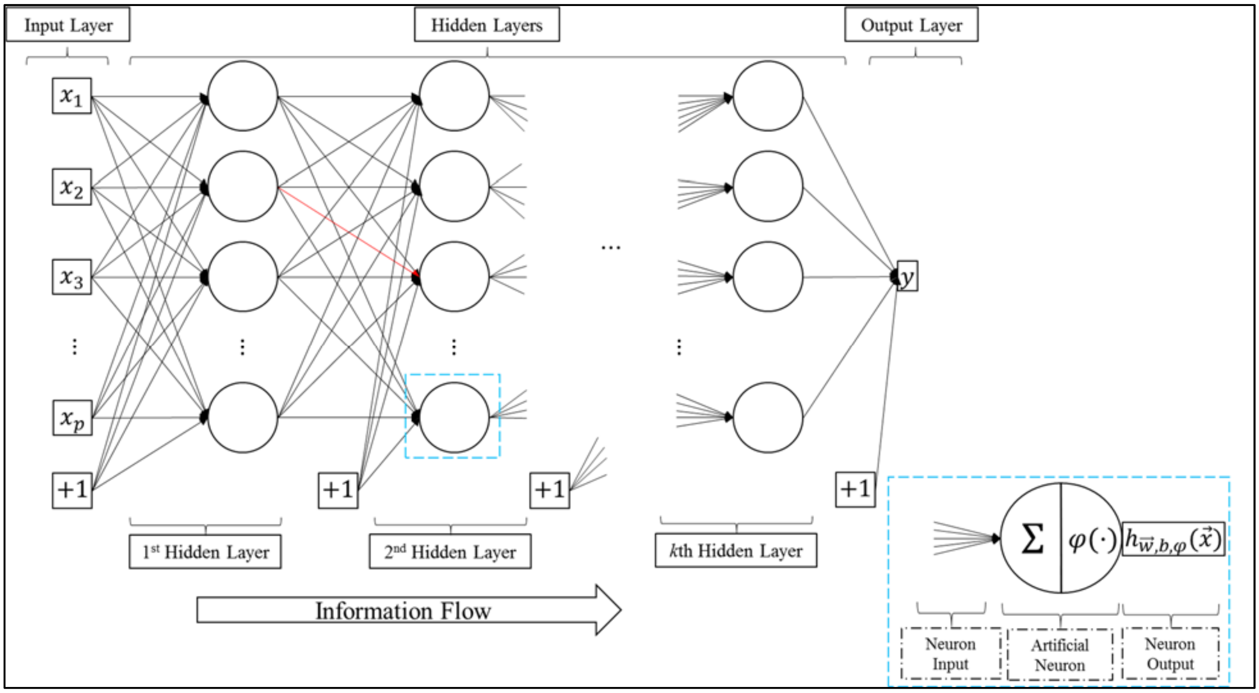


Figure 8. Feedforward Neural Network adapted from [103]

Within Figure 8, each black circle corresponds to an individual artificial neuron as depicted in Figure 7. The usage of the term network to describe NN imply that these algorithms are compositions of multiple functions connected together. The feedforward moniker is applied to these network as information flows in one direction only, from the input layer, through at least one intermediate layer of one or more artificial neurons, to the output layer. Additionally, FNNs do not contain any connections between non-consecutive layers, and every neuron in a preceding layer has a connection to every neuron in the next [95]. Thus, the FNN is easily represented as a directed acyclic graph as shown in Figure 8.

Within Figure 8, each solid black arrow represents a scalar weigh value applied to either the input data, or the output of a preceding neuron. Let n_l denote the number of layers in the network, also called the FNN's *depth*. In Figure 8 we have $n_l = k + 2$ for k-hidden layers with input-layer l_1 and output-layer l_{k+2}. We define $w_{i,j}^{(l)}$, as the weight associated with the connection between neuron j in layer l, from neuron i in layer $l + 1$. Using a similar convention, we define $b_i^{(l)}$ as the bias for neuron i in layer $l + 1$ [102]. For example examine the weight highlighted in red, for this weight we observe $w_{i,j}^{(l)} = w_{3,2}^{(2)}$ denoting the connection between the second neuron (as indexed from the top of the figure) in layer 2 with the third neuron in layer 3.

FNN are best thought of as function approximators. Within the FNN shown in Figure 8, the function to be approximated, is a mapping from the input data $\boldsymbol{x}$, to the output measure $\boldsymbol{y}$. This mapping is represented by the function $\mathbf{y} = f(\boldsymbol{x};\ \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the weights and biases within the network. During training (also called learning) the FNN is presented with example labeled training data. For each training example, $\vec{x}_i$, an accompanying target value, y_i, is also provided for the *i* = 1, 2, ..., *n* labeled training input-output pairs. As the FNN is provided with input-output data, the network adjusts the model weights and biases ($\boldsymbol{\theta}$) through *backpropagation*, resulting in the best function approximation [95].

The layers of artificial neurons between the input data layer and output layer are called *hidden layers*. The number of artificial neurons in a single hidden layer is called the layer's *width*. In Figure 7, we denoted the scalar-output of an artificial neuron as a function of one observation as $h_{\vec{w},b,\varphi}(\vec{x}_i)$. The output of an artificial neuron in a FNN is a function of the input data ($\vec{x}_i$) which is a vector corresponding the p-features of the data, the vector of scaling weights

($\vec{w}$), the bias term (b), and the selection of activation function (φ). Each artificial neuron converts a vector of input signals to a scalar output. Aggregating all the artificial neurons in a given hidden layer, l, allows us to define each hidden layer in a FNN as a vector-to-vector function, $f^{(l)}\left(\boldsymbol{x}^{*};\ \vec{\theta}\right)$**.** The number of features of the input data into layer l ($\boldsymbol{x}^{*}$) is dependent on the width (number of artificial neurons) in the preceding layer ($l-1$) and the dimension of the output vector is dependent on the width of layer l. In a FNN with at least one hidden layer, the overall approximation function is a composition of multiple vector-to-vector functions. For example, a single-hidden layer FNN would have the composition $\boldsymbol{y} = f(\boldsymbol{x};\ \boldsymbol{\theta}) = f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x};\ \boldsymbol{\theta})))$ with $f^{(1)}$ representing the input layer, $f^{(2)}$ the hidden layer and $f^{(3)}$ as the output layer. Likewise for a two-hidden layer FNN we would have the composition $\boldsymbol{y} = f(\boldsymbol{x};\ \boldsymbol{\theta}) = f^{(4)}(f^{(3)}(f^{(2)}(f^{(1)}(\boldsymbol{x};\ \boldsymbol{\theta}))))$, with layers $f^{(3)}$ and $f^{(2)}$as the hidden layers. The depth of a FNN is defined by the depth of the function composition, which is equivalent to the number of layers in the FNN [95]. The terminology of hidden implies that the behavior of these layers is not explicitly specified by the provided labeled training data. Instead, training the FNN through backpropagation, enables the network to select the appropriate vector-to-vector functions resulting in the best overall approximation function for the presented training data [95].

In Figure 8 we presented a FNN with a mapping from a vector described by a p-feature input observation to a single scalar output value, y. FNN are easily adaptable to higher dimensional output as shown in Figure 9.

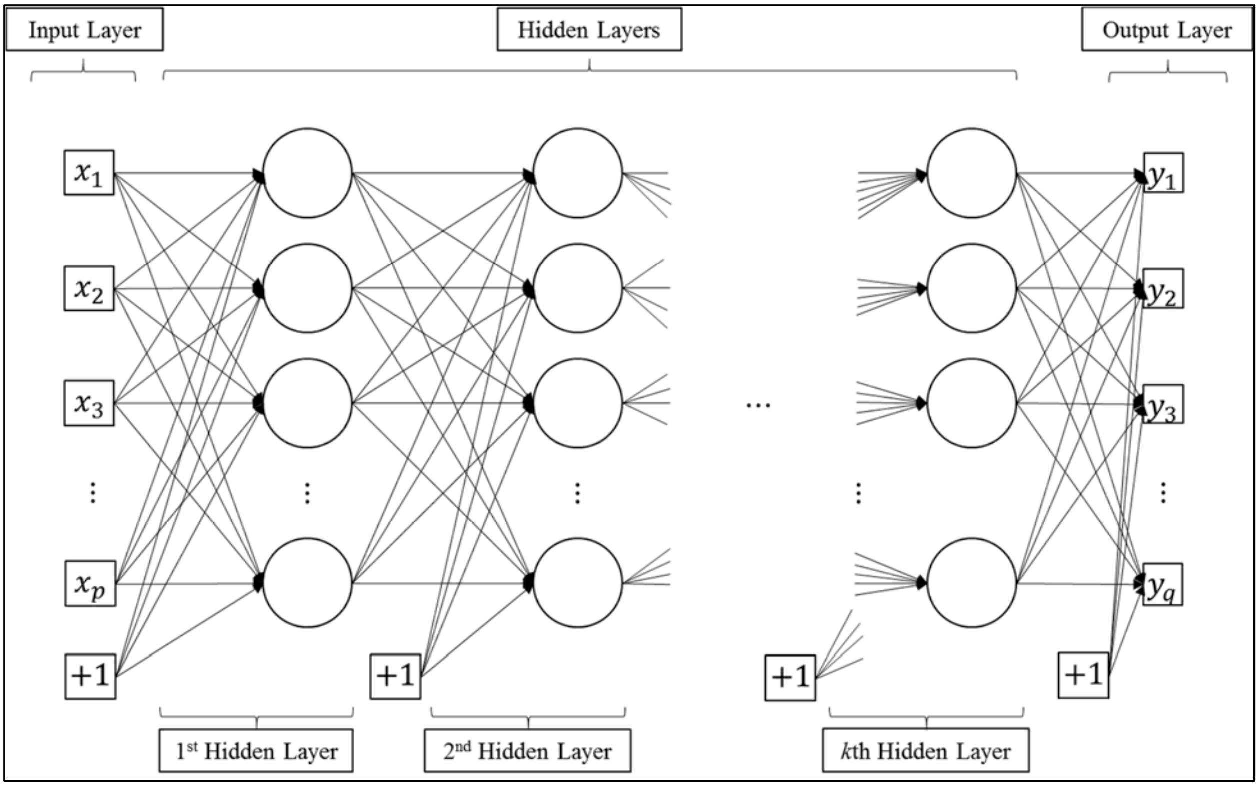


Figure 9. Multiple Response FNN adapted from [103]

In this figure we present a mapping $\mathbf{y} = f(\boldsymbol{x};\ \boldsymbol{\theta})$, where $\boldsymbol{x} \in \mathbb{R}^{n \text{ x } p}$ and $\boldsymbol{y} \in \mathbb{R}^{n \text{ x } q}$. There is no limit to the dimensionality of the input data ($\vec{x}_i \in \mathbb{R}^{1 \text{ x } p}$), nor the accompanying target value$s$ ($\vec{y}_i \in \mathbb{R}^{1 \text{ x q}}$) mathematically, although computational requirements for data processing (memory, processing time etc.) may become limiting factors for large datasets. In the multiple response FNN, backpropagation is utilized to adjust the FNN parameters to arrive at the best function approximation.

3.5.5 Backpropagation

Hecht-Nielsen [104] described backpropagation as “a new tool for approximating functions on the basis of examples.” Backpropagation is the process by which the changes to the FNN weights and biases ($\boldsymbol{\theta}$) are computed to improve the performance of the approximation function. Backpropagation is not in itself a learning algorithm, it is simply a method for computing the required changes to the each weight and bias in a FNN to improve the function approximation. The quality of the approximation capability of a FNN is evaluated using the *objective function* (also called *cost function*, *loss function*, *error function*) [95]. A common objective function utilized in FNN is the total error, E, defined as

$$\mathrm{E} = \frac{1}{2}\sum_{i=1}^{n}\sum_{m=1}^{q}\left(y_{i,m} - \hat{y}_{i,m}\right)^{2} \tag{1}$$

where i is an index over all n training examples, m is an index over all q output measures, $\hat{y}$ is the output from the FNN and y is the desired output provided in the dataset. Backpropagation seeks to minimize E through the use of gradient descent by computing the partial derivative of E with respect to each of weights and biases in the FNN.

Backpropagation proceeds in two phases, the forward-pass and then the backward-pass. Prior to training, initial weights and biases in the FNN are assigned as pseudo-random values. During the forward-pass the current weight and bias values are used to compute the FNN output values for each training example, after which E is calculated. The error is then differentiated with respect to each of the weight and bias terms in the FNN [105]. In the backward-pass, the weights and biases ($\boldsymbol{\theta}$) in the FNN are adjusted to reduce the total error, E as shown

$$\Delta\boldsymbol{\theta} = -\eta \frac{\partial E}{\partial \boldsymbol{\theta}} \tag{2}$$

where η is the *learning rate* dictating the magnitude in change to the weights. The learning rate parameter is often governed by the learning algorithm selcted by the analyst to update the weight and bias parameters in the NN, or can be manually set. Once the changes in the weights and bias updates are calculated, the FNN parameters are updated according to

$$\boldsymbol{\theta}_{\boldsymbol{t+1}} = \boldsymbol{\theta}_{\boldsymbol{t}} + \Delta\boldsymbol{\theta} \tag{3}$$

where $\boldsymbol{\theta}_{\boldsymbol{t}}$ corresponds to the parameter values during the current training period and $\boldsymbol{\theta}_{\boldsymbol{t+1}}$ are the parameters for the next training period. Backpropagation continues cycling through the forward and backward passess until a specified termination criteria, such as E being reduced to a predetermined value, is met [106]. A more detailed examination of backpropagation in FNNs can be found in [95], [104], [105], [107], [108].

3.5.6 Generalization

FNNs are useful only if they can accurately predict, or *generalize*, to novel data not previously observed by the network during training. To estimate the predictive capability of a FNN, we utilize a *test dataset* composed of labeled input-output pairs not used to adjust the weight and bias parameters of the network during training. The test dataset are samples independently and identically drawn from the same population as the training dataset. The capability of FNNs to predict accurately on the test data is the network's *generalization capability*, and the error associated with the test dataset is known as the *generalization error*. The goal of training FNNs is to reduce the generalization error to a minimum [95].

FNN training continues using the selected learning algorithm and backpropagation until a specified stopping criteria is reached. A common stopping criteria utilized in FNN research is to continue training until the generalization error is less than a specified value, or until the change of between successive iterations of the training algorithm, called *epochs*, is reduced below a threshold. Alternately, training may continue for a fixed number of epochs, or for a fixed amount of time. [95], [106], [109]. Figure 10 depicts a notional flow of the train and test process for NNs.

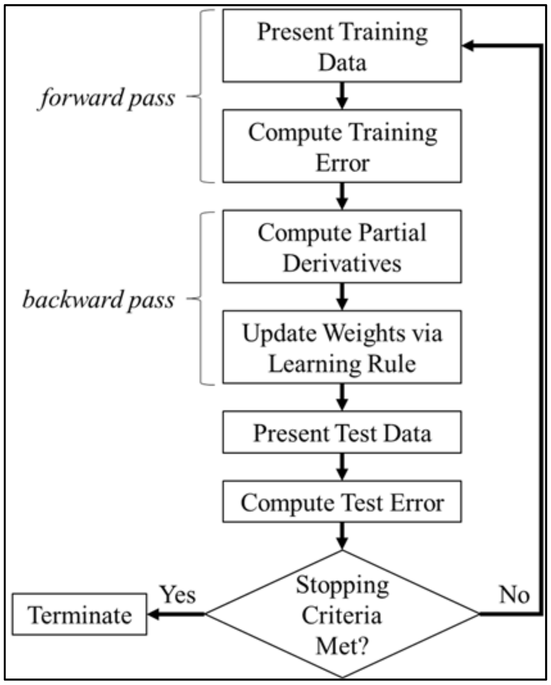


Figure 10. NN Train and Test Flow adapted from [106]

The training error will always decrease through subsequent iterations of the backpropagation procedure, however this is not the case for the test error. The training and test data are sample observations drawn from a larger population. FNN are trained to learn the population mapping from the input data ($\boldsymbol{x}$) to the output response ($\boldsymbol{y}$) using only the training data. There is a risk during training that the FNN will memorize the training data instead of learning the appropriate population function approximation. This occurrence is known as *overfitting*. Alternately, during training FNNs may fail to adequately learn the mapping from input data to output resulting in high training error [95]. This is known as *underfitting*. Figure 11 displays a typical plot of the test and training error observed during FNN training.

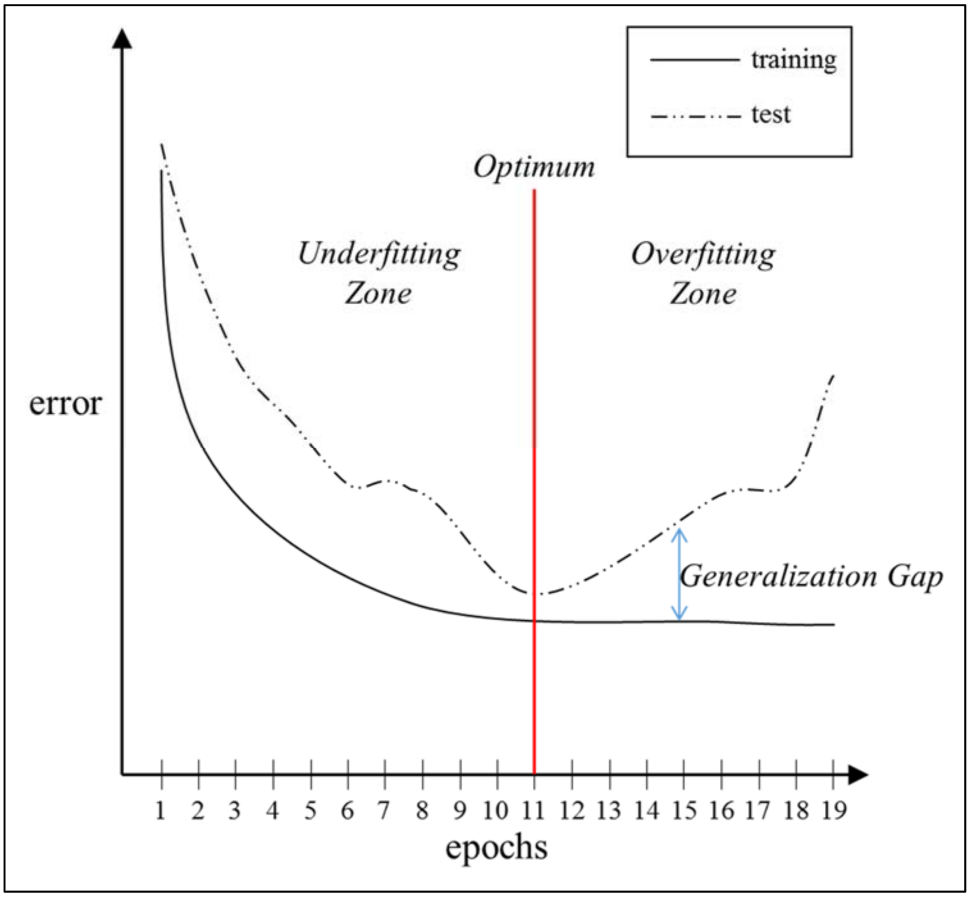


Figure 11. FNN Training and Test Error adapted from [95], [106]

Observe in Figure 11 that prior to epoch 11 the training error and test errors can both be reduced by additional training epochs. Premature termination of training before epoch 11 would result in underfitting. At epoch 11, the gap between the training and test errors, known as the *generalization gap*, are reduced to a minimum, indicating the weight and bias terms of the FNN are at the optimum values resulting in the best possible function approximation. Any additional training taking place after past epoch 11 results in a slight reduction of the training error at the cost of a rapid increase in the generalization error indicating that the FNN is no longer learning the population mapping, instead is memorizing the training dataset.

3.5.7 Activation Functions

The choice of activation function (φ) for the artificial neurons within FNN is a critically important consideration. Recall, the activation function is designed to mimic the all-or-nothing behavior of the biologic neuron, as such, it should suppress outputs from an artificial neuron when the cumulative inputs are small, and generate an output signal given sufficient inputs. We define the cumulative inputs of the weight-input sum product with the bias term as $z = b + \sum_x x_j w_j$ for j inputs to a neuron. Early work in the NN field considered the threshold function (Figure 12) as a potential artificial analog for the required biologic neuron behavior. The threshold function outputs a value of zero for inputs below a threshold value t, and outputs a value of one given inputs are greater than t. The threshold function is defined by

$$\varphi_{\text{Threshold}}(z, t) = \begin{cases} 0 & z < t \\ 1 & z \geq t \end{cases}. \qquad (4)$$

A plot of the threshold function for $t = 0.75$ is shown in Figure 12.

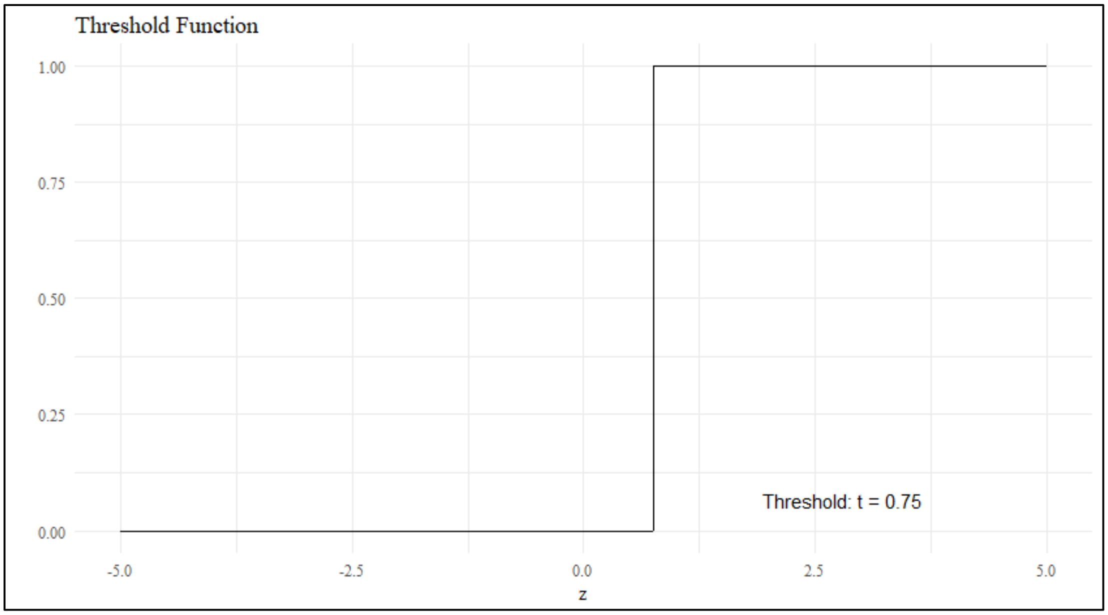


Figure 12. Threshold Function

As evident from Figure 12, the threshold function is not differentiable, consequently gradient-based parameter update methods, such as backpropagation, are not applicable to NNs using the threshold activation function. Weights and biases using the threshold function required manual setting by human operators. Subsequent NN work utilized a linear activation function defined by

$$\varphi_{\text{Linear}}(h) = h \tag{5}$$

whose derivative is given by

$$\varphi'_{\text{Linear}}(h) = 1. \tag{6}$$

While not exhibiting the all-or-nothing behavior of the biologic neuron, as the linear activation function is differentiable across the entire domain, these activation functions showed promise in regression type-analyses. As the linear activation function is differentiable, gradient descent methods, such as backpropagation, are available for adjusting the weight and bias parameters in FNNs with linear activation functions. The linear activation function's use in NN artechitecutres is a composition of many linear functions forming a linear function approximation for the NN. These linear models often have trouble learning highly non-linear functions, most famously failing to learn the *XOR* function [95].

To overcome the limitations of the linear activation function NN practitioners began to look at nonlinear activation functions such as the rectified linear unit (ReLU) function and the hyperbolic tangent (tanh) function. The rectified linear activation function is defined as

$$\varphi_{ReLU}(h) = \begin{cases} 0 & h \leq 0 \\ h & h > 0 \end{cases} \tag{7}$$

with derivative given by

$$\varphi'_{ReLU}(h) = \begin{cases} 0 & h \leq 0 \\ 1 & h > 0 \end{cases}. \tag{8}$$

The hyperbolic tangent function is defined as

$$\varphi_{tanh}(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}} \tag{9}$$

with derivative given by

$$\varphi'_{tanh}(h) = 1 - \varphi_{tanh}(h)^2. \tag{10}$$

A plot of the ReLU and hyperbolic tangent activation functions are depicted in Figure 13.

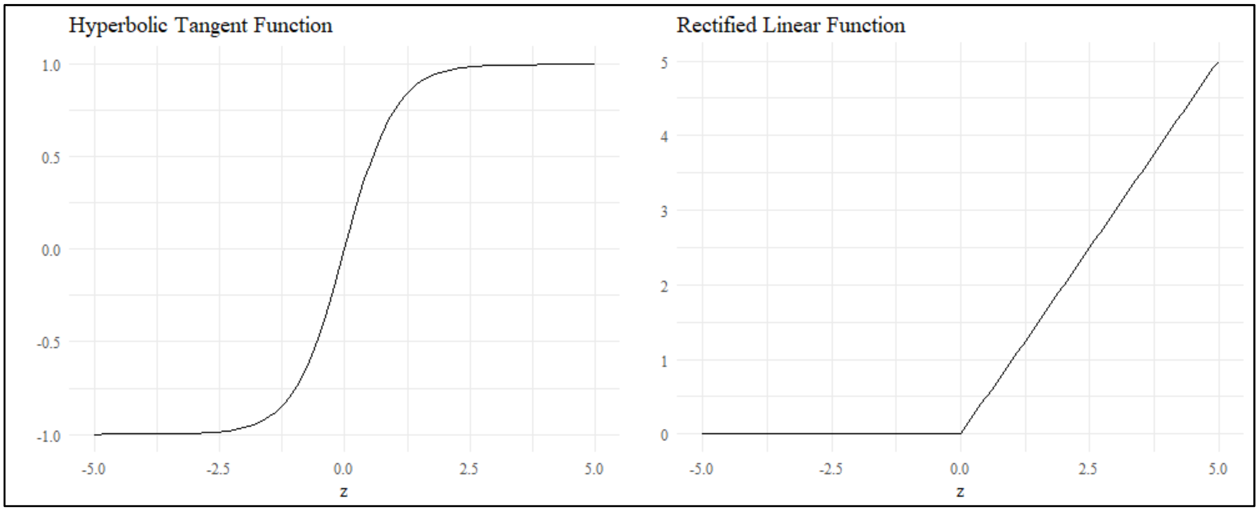


Figure 13. Hyperbolic Tangent and Rectified Linear Unit Functions

The hyperbolic tangent closely resembles the threshold function shown in Figure 12, however the function is differentiable across its domain, consequently, gradient based parameter update methods are available for parameter training. Furthermore, as this is a nonlinear function, FNNs using the hyperbolic tangent function are able to learn both linear and highly nonlinear function approximations [95]. The hyperbolic tangent function is an example of a class of functions known as squashing functions. Hornik et al. proved that given a FNN with a least one hidden layer using neurons using a squashing function, are able to approximate any Borel measurable function to any desire accuracy provided a sufficient number of hidden artificial neurons are present [110].

The ReLU activation function is a modification to the linear activation function in which the ReLU returns zero when the cumulative input is less than zero. The ReLU is nearly differentiable across the entire domain with a discontinuity at $h = 0$ with the added benefit of a computationally simple derivative, where defined. As the ReLU is a nonlinear function, it is able to approximate both linear and highly-nonlinear functions when used in NNs. As the ReLU function is not differentiable across its entire domain, we can conclude that it is a nonpolynomial function. Leshno et al. extended the work of Hornik et al. and proved a more general result, given a FNN with a least one hidden layer using neurons with a nonpolynomial activation function, these networks are able to approximate any Borel measurable function to any desired accuracy provided a sufficient number of hidden artificial neurons are present [111].

Leshno et al, and Hornik et al. conclude that any failure of a FNN to adequately learn a function approximation to the desired degree of accuracy can be attributed to insufficient learning, an insufficient number of hidden neurons, or a stochastic (probabilistic) relationship between the input-output data. Hornik et al. and Leshno et al. consider FNNs as universal function approximators, however they note, the number of artificial neurons required to meet the desired degree of accuracy (assuming a non-stochastic mapping between the input-output features) is a question still to be answered [110], [111].

3.5.8 Hyperparameters

We have briefly discussed NNs and described the method underpinning all learning processes. The specific manner in which NNs operate are controlled by model parameters known as hyperparameters. The hyperparameters differ from the weight and bias parameters as the hyperparameters are traditionally not adjusted during training, they are set prior to learning activities and remain unchanged during the course of learning. Hyperparameters control the behavior of the NN algorithm and hyperparameter setting values can have varying effects on the final trained NN model as well as the overall performance of the function approximation. [95], [112], [113]. The *no free lunch theorem* prescribes that no machine learning algorithm (to include NNs) is always universally superior to any other [95], [114]. Consequently, despite the universal function approximation capabilities of NNs with appropriate activation functions, NN performance is dependent on the appropriate selection of hyperparameters controlling the algorithms learning behavior.

We have previously discussed some common NN hyperparameters, such as the activation function, number of hidden layers, number of neurons per layer, choice of objective function, etc. We now describe the hyperparameters evaluated in this work.

3.5.8.1 Depth and Width

The universal approximation capability of NNs with one hidden layer using the appropriate activation functions and a sufficient number of neurons was previously discussed in section 3.5.3. It may require an exponential number of hidden neurons to approximate a function to the desired degree of accuracy with a NN with only one hidden layer. Recent work in the NN field empirically shows that superior results are obtained with deeper NNs in a variety of fields [115]–[125]. The use of deeper architectures enables the NN to learn intermediate representations of the data within the hidden layers during training. These learned intermediate representations may be more advantageous to the function approximation desired for the NN [95].

The number of hidden layers and the number of artificial neurons in each layer are evaluated. These values are required to be integer values greater than one and we restrict the number of neurons in any layer to be less than the number of features in the input layer of the NN. As the depth of the NN architecture, and the number of artificial neurons increase, the training time is expected to grow. Additionally, the number of neurons per layer and the depth of the NN are hierarchically related, with only two-hidden layers there cannot be any artificial neurons in the nonexistent third hidden layer.

3.5.8.2 Dropout

Dropout is a regularization method for preventing overfitting in NNs. Dropout is closely related to *bagging* in which independent NN models, each created from different set of training and test data, are averaged together in a final ensemble model [126]. Bagging is a computationally expensive process for large NN architectures, dropout is a computationally inexpensive method capable of scaling to large NNs with an exponentially large number of NN ensembles [95].

During each training pass through the data with dropout, each input layer and hidden layer neuron are randomly and independently assigned a masking probability, called the *input dropout rate* and the *hidden dropout rate* respectively. If the masking probability is active the output from those neurons is suppressed and training continues without the affected neurons contribution to the NN. Subsequently, the NN is forced to adapt to the removal of the masked information using the information from non-masked neurons in the network. Through subsequent dropout training iterations of the NN, a large number of different NN structures are evaluated and their results averaged together into a final ensemble mode. The dropout ensemble models are not independent, each subsequent network shares information gained from previous NN generations. Empirical results with NNs trained with dropout regularization indicate that dropout is a superior regularization method compared to others in preventing overfitting and improving regularization [95], [127].

Dropout, when employed, requires the specification of the input and hidden dropout rates hyperparameters prior to training. These values are real-valued numbers greater than zero and less than one. Differing values of the input and hidden dropout rates are evaluated in this work.

3.5.8.3 Learning Rate

The learning rate hyperparameter was previously discussed in section 3.5.5. This parameter controls the size of the change in the weights and biases parameters updated during training to minimize the objective function. Modification to the learning rate controls how quickly the NN converges to an optimum value. Large learning rates may cause the objective function, and thus total error, to increase during training, and small learning rate values will result in slow learning [128].

Often during training the learning rate set at the start of training is too large once the training error begins to approach its minimum value. *Rate annealing* is a method for gradually decreasing the learning rate as the NN proceeds through training. The learning rate parameter is decreased according to

$$\eta_t = \frac{\eta}{1 + (N * \eta_{\text{annealing}})} \qquad (11)$$

where η_t is the updated learning rate of the tth iteration of training, η is the learning rate, N is the number of training samples and $\eta_{\text{annealing}}$ is the annealing rate, a hyperparameter to be specified [109], [129].

In NNs with deep architectures a common problem observed is that the weight and bias gradient of layers closer to the input layer of the network are often much smaller than the gradient of the weight and bias parameters closer to the output layers. This phenomena is known as the *vanishing gradient problem*. Consequently, as the number of layers of a NN, increase the gradient updates back-propagated to the layers closest to the input layer grow small, slowing the training of the weights and biases in these layers resulting in slow training of the NN overall. A method to correct for the vanishing gradient problem is the introduction of a *rate decay* hyperparameter which decreases the learning rate of layers later in the network, promoting increased weight and bias updates in earlier layers. The change of the learning rates as a function of the layer is given by

$$\eta(L) = \eta * (\eta_{\text{decay}})^{L-1} \tag{12}$$

where $\eta(L)$ is the learning rate of layer L, and η_{decay} is the rate decay [109], [129].

The learning rate, rate annealing, and rate decay are hyperparameters evaluated in this work. The learning rate requires a real-value number greater than one, whereas the rate annealing and rate decay values require real-values greater than or equal to zero. Note, if the rate annealing or rate decay values are set to a value of zero, this implies that these leaning enhancements are not applied to during NN training.

3.5.8.4 Momentum

Momentum is a method to improve the training speed of NNs. Momentum enables prior weight and bias parameter updates to influence future updates in the network. Parameter updates resulting in a large change to the objective function in previous updates are retained in subsequent updates to prevent the training algorithm from becoming stuck in local minimums [109], [130], [131]. Momentum uses a *velocity* term defined by

$$\boldsymbol{v}_{t+1} = \mu \boldsymbol{v}_t - \eta \frac{\partial E}{\partial \boldsymbol{\theta_t}} \tag{13}$$

where μ is the momentum hyperparameter, $\boldsymbol{v}_t$ is the current velocity, η the learning rate, and $\frac{\partial E}{\partial \boldsymbol{\theta}_t}$ is the gradient of the parameters at the current iteration. The velocity modifies the future parameter updates by

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} + \boldsymbol{v}_{t+1}. \tag{14}$$

NN training often proceeds quickly in early iterations of the learning algorithm and subsequently begins to slow as local minima of the objective are encountered, therefore momentum early in training is less important than momentum later. Momentum is often increased to a maximum specified value according to the number of training iterations completed. The momentum procedure is defined by three hyperparameters, *momentum start*, *momentum ramp*, and *momentum stable.* The momentum start is the initial momentum, the momentum stable is the final momentum value. The momentum ramp is the number training examples over which the momentum start values is increased to the momentum stable value [109].

The momentum start, stable, and ramp values are all hyperparameter values considered in this work. The momentum start and stable values are required to be real values greater than or equal to zero, and the momentum ramp values is required to be a positive integer value. Values of momentum stable of zero indicate that the momentum will remain constant over the course of training and when both momentum start and stable are both zero, momentum will not be a method utilized in training.

3.5.8.5 Nesterov Accelerated Gradient

Nesterov accelerated gradient (NAG) is a modification to the classic momentum algorithm. NAG incorporates the predicted gradient change of the NN parameters in the next iteration of training, using first order gradient information in the current iteration. This modification increases the learning rate for current iterations of training if the predicted future gradient is large, and decreases the learning rate if the future gradient is predicted to be small [129]. NAG modifies the velocity equation of momentum adding in an approximation for the future gradients as shown

$$\boldsymbol{v}_{t+1} = \mu \boldsymbol{v}_t - \eta \frac{\partial}{\partial \boldsymbol{\theta}} [\boldsymbol{\theta}_{\boldsymbol{t}} + \mu \boldsymbol{v}_t]. \tag{15}$$

Subsequent NN parameter updates are computed using the traditional momentum update equation as shown in Equation (14) [131].

The use of NAG is a hyperparameter studied in this work.

3.5.8.6 Adaptive Learning

Manual configuration of the six hyperparameter settings required for the learning rate and momentum training methods typically requires a tuning process by which subsequent iterations of hyperparameters are tested against one another until the NN meets adequate performance. Zeiler [128] devised an *adaptive learning* procedure in which the search for the best learning rate and momentum hyperparameters is replaced by an algorithm called *ADADELTA.* ADADELTA uses first order information to find appropriate learning rates on a per-feature basis. The Zeiler adaptive learning algorithm requires the specification of two hyperparameters, *epsilon* and *rho*. The epsilon hyperparameter is an adaptive form of the rate annealing hyperparameter from the learning rate methodology, and the rho hyperparameter is similar to momentum and acts as a memory of past parameter updates for subsequent iterations [109]. ADADELTA has empirically demonstrated robustness to large gradients during training and large noise in the dataset. Furthermore ADADELTA is adaptive to a wide variety of NN sizes. Specifics regarding the ADADELTA algorithm can be found in [128].

Epsilon and rho are hyperparameters evaluated in this work. When ADAGRAD is utilized as a learning rule, the six hyperparameters of learning rate and momentum are not specified, however, ADAGRAD can use NAG to accelerate training.

3.5.8.7 Activation Function

Activation functions are discussed in section 3.5.7. The choice of ReLU or hyperbolic tangent activation functions is a hyperparameter evaluated in this work.

3.5.8.8 Parameter Norm Penalties

Parameter norm penalties are a common form of regularization for NNs implemented to prevent overfitting to training data. These methods work by reducing the capacity of the NN, limiting the efficacy of the learned function approximation. The NN capacity is limited through the introduction of a parameter norm penalty to the objective function as shown

$$\mathrm{E}_P = \mathrm{E} + \alpha\Omega(\boldsymbol{\theta}) \tag{16}$$

where E_P is the penalized objective function, E is the total error from Equation (1), $\Omega(\boldsymbol{\theta})$ is a function of the NN weights and biases, and α is the norm penalty hyperparameter [95].

The most widely implemented NN parameter norm penalty is $\mathbf{L}^2$ regularization (also called *weight decay, ridge regression*, or *Tikhonov regularization)* where the penalty function is defined as

$$\Omega_{\mathrm{L2}}(\boldsymbol{\theta}) = \alpha_{\mathrm{L2}} \frac{1}{2} \|\boldsymbol{\theta}\|_2^2 \tag{17}$$

where α_{L2} is the hyperparameter. $\mathbf{L}^2$ regularization drives the weights and biases in the NN toward zero. Considering the combined effect of the error and $\mathbf{L}^2$ regularization, the only parameters retained in the final NN are those associated with significantly gradient contributions resulting in improvement to the objective function value. Parameters associated with gradient directions along which the overall objective function will not significantly decrease, are decayed toward zero [95].

$\mathbf{L}^1$ regularization is used to impose *sparsity* on the final NN. Sparsity in this context being values of the NN parameters with an optimal value of zero. The $\mathbf{L}^1$ penalty function is defined as

$$\Omega_{\mathrm{L1}}(\boldsymbol{\theta}) = \alpha_{\mathrm{L1}} \frac{1}{2} \|\boldsymbol{\theta}\|_1. \tag{18}$$

where α_{L1} is the corresponding hyperparameter [95].

The $\mathbf{L^1}$ and $\mathbf{L^2}$ regularization hyperparameters (α_{L1} and α_{L2}) are tested in this work. These values are required to be greater than or equal to zero, with larger hyperparameter values corresponding to a larger regularization effect. Hyperparameter values of zero correspond to the removal of the regularization effect from the objective function.

3.5.8.9 Initial Weight Distribution and Data Scaling

The initial weight distribution describes the manner in which the random initial weight and bias assignments are made in the NN. The scale of the features within the test and training data describes the range of each feature within the data. The interplay between the values of the initial parameters, the scale of each feature, and the activation function is an important consideration.

Given the hyperbolic tangent activation function, the combined effect of the initial parameter weights and scale of each feature should fall in the region where the hyperbolic tangent is approximately linear, [-1, 1]. If the weights are too large or too small or the data is scaled inappropriately, then the input into the activation functions will occur where the hyperbolic tangent has small gradients, resulting in slow initial training. Assuming that the initial parameter values and data scale result in activations occurring in the linear region of the hyperbolic tangent activation function, learning will proceed quickly during the early training iterations [130].

The ReLU activation function has a derivative of zero for inputs less than zero, and a constant derivative for inputs greater than zero. When using the ReLU activation function it is recommended that the effect of the initial weights and data scaling result in positive inputs to the activation function ensuring that initial training progresses quickly. A common method is to ensure initial inputs to the ReLU activation function are positive is to ensure that the bias units in the NN are all initialized with a value of at least 0.10 [95], [130].

There are three common methods for NN weight initialization, drawing from a *uniform distribution* on the [-1, 1] interval, drawing from a *normal distribution* with mean zero and standard deviation of one, and drawing from an *adaptive uniform distribution* where the weight initiation distribution is based on the number of weight and bias parameters in the NN [109]. Common methods for scaling feature within the test and training data are to scale each feature with the datasets to values of [-1, 1], [-0.5, 0.5] or [0, 1]. The parameter initialization method selected and the feature scaling are hyperparameters tested in this work.

3.5.8.10 Shuffle Training Data

NNs learn most quickly when presented with novel data instances, therefore it is preferable to continually train the networks with the most unfamiliar data [130]. We are currently unaware of any method for determining real-time what observations from the training data are the most novel. A simple heuristic to increase the probability that novel data is presented to NNs during training is to randomize the training data presented.

Randomization of the order of the training data presented to the NN during training is a hyperparameter evaluated in this work.

3.5.8.11 Minibatch Size

The minibatch size describes the number of training exemplars presented to the NN before updates to the weight and bias parameters are made during training. Training methods which update NN parameters after each individual training example are known as *on-line* training methods. On-line training methods from a noisy approximation of the gradient direction at the current NN parameter values. Training in which parameter changes are accumulated over the entire training dataset are known as *batch* training methods. Batch training methods are able to calculate the direction of the true gradient direction at the current NN parameter values. *Minibatch* training requires the specification of the number of training examples to evaluate before parameter updates are made. Minibatch training methods use a less noisy approximation of the true gradient method than on-line methods, and as the size of the minibatch grows the approximated gradient direction approaches the true gradient direction [132].

The minibatch size is a hyperparameter evaluated in this work. The minibatch size is required to be an integer value between on and the number of training observations. A minibatch size of one implies on-line learning, and a minibatch size equal to the number of training examples implies batch learning.

3.5.8.12 Average Activation

The *average activation* imposes a sparsity constraint on the hidden layer neurons. This sparsity constraint prevents the NN from learning an exact function approximation to the data, requiring the NN to learn novel ways of representing data within the hidden layers. The average activation increases the bias on neurons with large activations and decreases the bias on neurons with small activations [102]. Recall the activation of each neurons in the NN can be represented by

$$h = \varphi\left(\sum \vec{w}\vec{x} + b\right) \tag{19}$$

where $\vec{w}$ is the vector of weights acting on the $\vec{x}$ data into the neuron, b is the bias, and φ is the neuron's activation function. When average activation sparsity is employed the bias terms for each neuron are updated according to the following

$$b_{t+1} = b_t - \eta\beta(h_t - \rho) \tag{20}$$

Where b_t is the current bias value, b_{t+1} is the updated bias value, η is the learning rate, β is the sparsity learning rate hyperparameter, h_t is the current activation of the neuron and ρ is the desired average activation of each neuron (a hyperparameter). The bias adjustments are made after the backward pass in backpropagation [102].

Average activation with the sparsity learning rate (β) and desired average activation (ρ) are hyperparameters evaluated in this work. The sparsity learning rate is required to be a real number greater than or equal to zero, and the average activation is required to be a real number. The desired average activation hyperparameter is tied to the choice of activation function which dictates the possible range of output values of each neuron.

3.5.8.13 Max W^2

The max W^2 hyperparameter imposes an upper limit to the sum of the square of the incoming weights values into each neuron in the network. This is similar to the $\mathbf{L^2}$ regularization discussed in section 3.5.8.8, however the max W^2 imposes an upper limit to each individual neuron and this hyperparameter is not a part of the NN objective function. Limiting the incoming weights is especially useful when the unbounded ReLU activation function is utilized in the NN to prevent weights from growing exponentially.

Max W^2 is a hyperparameter evaluated in this work. The max W^2 value is required to be a real number greater than zero.

3.6 Autoencoders

Goodfellow et al. [95] describes an autoencoder as "The quintessential example of a representation learning algorithm." Autoencoders, (also known as autoassociative NNs or replicator NNs) are a type of FNN designed to perform an identity mapping of the input data. In other words, ANNs copy the input data to the output data. The ANN is composed of two parts, the *encoder function* mapping the input data into a representation internal to the NN, or *code* and the *decoder function* mapping from the code space back to the original. Let $\boldsymbol{h} \in \mathbb{R}^{n \times s}$ represent the code representation internal to the NN, the function $\boldsymbol{h} = f_{Encode}(\boldsymbol{x}, \boldsymbol{\theta})$ is the mapping from the natural data space ($\boldsymbol{x} \in \mathbb{R}^{n \times p}$) to the code space ($\boldsymbol{h} \in \mathbb{R}^{n \times s}$), and the function $\boldsymbol{x} = f_{Decode}(\boldsymbol{h}, \boldsymbol{\theta})$ is the mapping back to the natural space from the coded internal representation [95].

ANNs with the capacity to learn an exact identity mapping without error are not very useful as there would be no modification to data already available for analysis. Autoencoders are designed with constraints restricting their capability to approximate the identity function. By limiting their capacity, the autoencoder is forced to prioritize which aspects of the data set to learn to reproduce, in doing so, also learning properties of the data of interest to analysts. The efficacy of autoencoders as a data analysis tool are dependent on the constraints placed limiting their capability to exactly reproduce the data [95], [133].

The *undercomplete autoencoder neural network* (UANN) restricts the networks capacity to learn an exact identity mapping by imposing a restriction on the dimensionality of the code representation of the data. The restriction is obtained through the introduction of a *bottleneck* *layer* in the NN as shown in Figure 14. The bottleneck layer has fewer neurons than the input/output data, consequently the internal code representation of the data $\boldsymbol{h} \in \mathbb{R}^{n \times s}$ has a smaller dimension than the input data, $s < p$. The internal dimensional reduction of the bottleneck layer forces the NN recreate higher dimensional output data from the compressed representation. In order for the ANN to minimize the resultant error during training, it must learn the most salient features of the data to recreate the input data at the output layer.

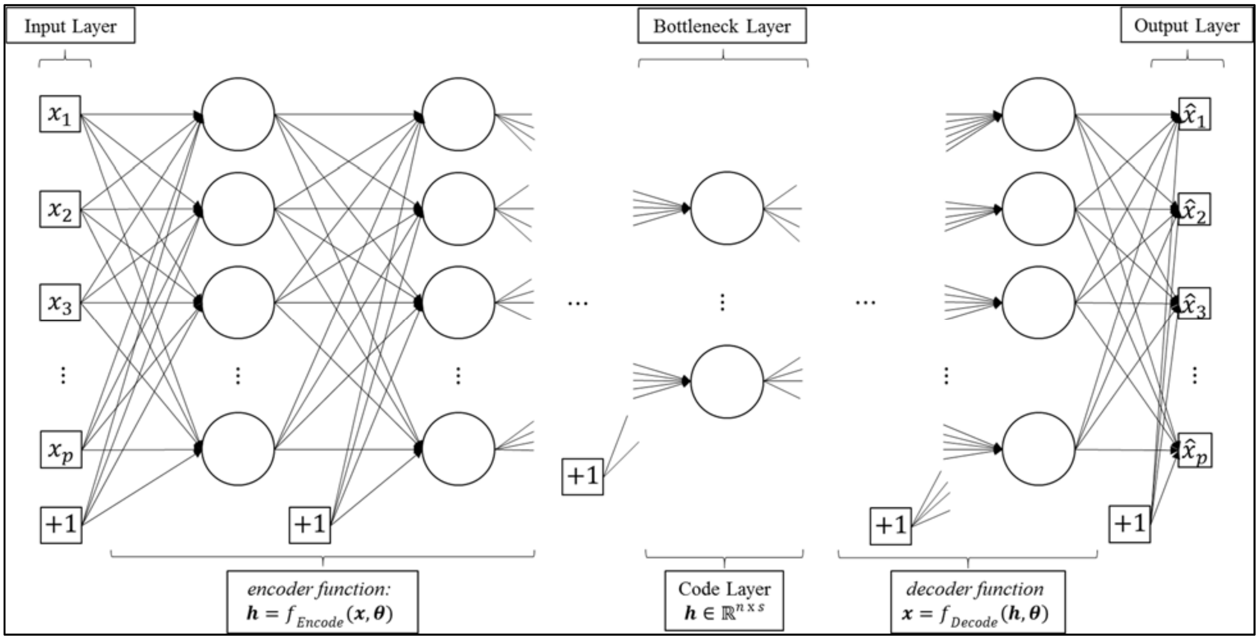

Input Layer
Bottleneck Layer
Output Layer
x_1
x_2
x_3
x_p
+1
+1
+1
+1
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_p$
encoder function:
$\boldsymbol{h} = f_{Encode}(\boldsymbol{x}, \boldsymbol{\theta})$
Code Layer
$\boldsymbol{h} \in \mathbb{R}^{n \times s}$
decoder function
$\boldsymbol{x} = f_{Decode}(\boldsymbol{h}, \boldsymbol{\theta})$

Figure 14. Undercomplete Autoencoder Nerual Network adapted from [103]

The NN in essence compresses redundant information in the bottleneck layer, only retaining the patterns in the data useful for differentiation of non-redundant information [52], [95], [133]. As the capacity of the autoencoder is limited by the bottleneck layer, so too is the capability of the network to retain information within the data used for differentiation of non-redundant information. Consequently, the ANN will reproduce common data with less error than rare (anomalous) data [56].

Autoencoders have seen wide adoption in data reduction tasks as well as information retrieval tasks. In addition, autoencoders have also shown promise as a data preprocessing tool in classification tasks such as computer image recognition [134]. Japkowicz et al. [52] was the first to utilize ANN for the task of anomaly detection. Their research utilized supervised data sets for anomaly detection. In the Japkowicz et al. approach, a semi-supervised data subset composed entirely of non-anomalous observations was used to train the autoencoder. Once trained to the normal data, the autoencoder was evaluated against the anomalous observations and compared to other common anomaly detection methods. Compared to the traditional anomaly detection methods, the autoencoder method performed better than or equal the other methods examined in three different domain areas [52].

Hawkins et al. first used ANN to detect anomalies in an unsupervised dataset. The datasets contained class labels, however during NN training, these labels were omitted from consideration. The labels were only used to evaluate the performance of the NNs after training. In the Hawkins et al. work, the datasets studied were first split into two distinct subsets, a training and testing set of data. The training set being used for parameter updates, and the test set for evaluating performance of the NN. The NN objective function used in Hawkins et al. was the mean square error defined as

$$\mathrm{MSE} = \frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{i,j} - \hat{x}_{i,j})^2 \qquad (21)$$

where p is the number of features, n is the number of observations, $x_{i,j}$ is the true value of the ith observation for the jth feature, and $\hat{x}_{i,j}$ is the ANN predicted value.

In addition to using unsupervised datasets, the Hawkins et al. work introduced the concept of using a score based method, known as the *outlier factor* (OF), for identifying outliers in datasets analyzed with autoencoders. The outlier factor for each observation in the dataset is the average reconstruction error over all features defined as

$$\mathrm{OF}_i = \frac{1}{p}\sum_{j=1}^{p}\left(x_{i,j} - \hat{x}_{i,j}\right)^2 \tag{22}$$

where p is the number of features, $x_{i,j}$ is the true value of the ith observation for the jth feature, and $\hat{x}_{i,j}$ is the ANN predicted value. The OF is evaluated for all observations in the dataset using a trained autoencoder, with higher OF values considered to be more likely anomalous data [56]. The Hawkins et al. work, is the earliest found example of a purely unsupervised anomaly detection method using ANNs. The autoencoder methods established in Hawkins et al. were tested against traditional anomaly detection methods in Williams et al. using four datasets, one of which was in the domain of network intrusion detection. In the Williams et al. work, the ANN methods performed comparably to the traditionally anomaly methods tested and in the case of network intrusion detection, the ANN method's performance surpassed the alternate traditional anomaly detection methods [87].

3.7 Hyperparameter Design of Experiments

Identification of the best hyperparameter settings for a NN is a non-trivial task. Generally, the performance of multiple NNs with different hyperparameters are compared to one another using the generalization error of the test dataset as a metric. The NN with the smallest generalization error is declared the best, and those hyperparameters are selected as the optimum for the particular problem and dataset analyzed. Also of interest to NN practitioners are the computer resource required as well as the time required for NN training, both of which are affected by the hyperparameters, but are also highly depended on the computer system used to perform training [95]. Four basic approaches are employed for hyperparameter optimization, manual, grid, random, and model-based.

Manual hyperparameter selection is perhaps the most widely utilized optimization scheme. In manual optimization, NN subject matter experts manually select hyperparameter values based on prior experience and run the NNs examining the generalization error for each NN configuration. The NN subject matter experts continue to adjust the NN hyperparameters until a suitable hyperparameter set is found in a serial trial-and-error process. Manual hyperparameter selection can be an efficient process as the subject matter experts can quickly diagnose deviations from the expected NN performance and make the appropriate corrections [135]. The use of manual hyperparameter search is ill suited for NN novices, and limits the reproducibility of results across differing datasets and research domains, however manual search does give the NN subject matter experts insight into the effects of hyperparameters on the learning process. Furthermore, manual search is a technically simple process and requires no additional resources beyond those required to execute NN training [136].

The methods of grid and random hyperparameter selection are similar processes. Grid search involves the selection of a small finite set of values for each of the hyperparameters of interest. Then, a NN is trained for each Cartesian product of the set of values for each hyperparameter. In the DOE literature this is known as a full factorial experiment. Each combination of hyperparameter values is tested and the combination of hyperparameters with the best generalization error is selected. In random hyperparameter search a marginal distribution for each hyperparameter of interest is defined. Then hyperparameter values are randomly drawn from the distributions, and a NN with those hyperparameter values is trained. Random search of the hyperparameter space continues until a suitable set of hyperparameters is found, or a predefined search time expires. Grid searches typically only allow a small number of hyperparameters at a few levels to be explored due to the curse of dimensionality, as the number of hyperparameters and the number of values for each hyperparameter increase, the total number of potential NNs grows exponentially [95]. Random search methods are more efficient than grid search in both the number of hyperparameters explored and have demonstrated superior performance in selection of the hyperparameter values resulting in decreased generalization error [136].

The selection of hyperparameters for a NN can be viewed as an optimization problem in which we seek a set of hyperparameters, $\mathbf{\Lambda}$, resulting in the smallest generalization error. In model hyperparameter selection we construct a model approximation of the generalization error as a function of the hyperparameters and use gradient-based optimization methods to determine the optimal set of hyperparameter values for use in the NN [95]. There are numerous methods to construct the hyperparameter-generalization model as discussed in [137]. Numerous authors have used DOE as a method for determining the optimal hyperparameters for NNs in a variety of domain areas as documented in [138]–[144], [145, Ch. 19], [146].

DOE is the scientific process of planning, executing, and analyzing experiments to ensure that the appropriate data is collected to draw objectively valid conclusions regarding the underlying process of study. A designed experiment is a series of individual *tests* in which purposeful changes are made to *factors* to observe the effect on the measured *response*. A factor is an independent variable with different *levels* which the experimenter changes to elicit a change in the response. What distinguishes designed experiments from traditional experiments is in a designed experiment the individual tests are selected strategically so when analyzed collectively as an experiment, the experimenter gains the greatest amount of information regarding the relationship between the factors and response, in the minimum number of experimental runs [147].

In the context of NN hyperparameter optimization, the process of study we are interested in is the relationship between the change in hyperparameters and the resultant generalization error of the NN. The factors are the hyperparameters, the levels are the values of each hyperparameter, and the response is the generalization error. A test is the evaluation of one NN with the experimental prescribed hyperparameter settings. The hyperparameter designed experiment is the set of all NNs evaluated.

The overall goal of a designed experiment is to fit a statistical model analytically describing the relationship between the factors and the response. DOE utilizes *analysis of variance* (ANOVA) to determine the statistical model. ANOVA enables the experimenter to attribute changes in the response to individual changes to the levels of a factor. Once a suitable analytic model relating the factors to the response is found, gradient-based optimization are applied and the hyperparameter levels resulting in the lowest generalization error are estimated. As the derived analytic model is only an approximation of the true function relating the hyperparameters to the generalization error, there is uncertainty around the estimated hyperparameter levels, validation testing is strongly advised. Additional information regarding ANOVA can be found in [147], [148].

The core principal underpinning DOE methodology is the development of a sound experiment. When constructing an experiment, the experimenter implicitly specifies the potential forms of the final statistical model relating the factors to the response. Complex statistical models with high-order interactions and nonlinear effects require experiments with a larger number tests, whereas simple statistical models can be obtained with smaller experiments. Classic DOE methods advocate for a sequential experimental process, in which simple smaller experiments, called screening experiments, are first conducted to determine if factors affect the response. Screening experiments enable the experimenter to remove factors that do have a statistically and/or practical effect on the response thereby reducing the size of subsequent experiments. Once insignificant factors are removed, follow-on experiments are used to determine the appropriate statistical model relating the retained significant factors to the response. Removing the insignificant factors enables the experimenter to disregard regions of the experimental space where factors have no discernable impact on the response resulting in more efficient experiments [147]. Additional information regarding DOE can be found in [147].

IV. Methodology

4.1 Chapter Overview

In this chapter we present the methodology employed to derive the ANN anomaly detector for the IDPS log file dataset. We begin with a brief exploratory analysis of the IDPS log dataset to select appropriate features from that dataset for use in UANN analysis. Once the appropriate features are identified, the reduced dataset is adapted for us in a NN and split into three distinct training and testing subsets for NN evaluation. Screening designed experiments are employed to identify the categorical hyperparameter levels significant for minimization of the generalization error. Once the categorical hyperparameter levels are identified we create additional designed experiments to optimize the numeric hyperparameter values minimizing the generalization error across the three training-testing subsets using statistical models. The hyperparameter values are validated by comparing the statistical model predicted generalization error to the actual generalization error obtained using UANNs with the statistical model predicted optimum hyperparameter values. After identifying a validated set of optimum hyperparameter values, we train an UANN with those values on the full dataset and calculate the outlier factor score for each observation. We display the results graphically for subsequent computer security expert analysis.

For the data preparation this work utilized the R programming language [149] and the RStudio integrated development environment (IDE) [150]. We use JMP PRO software to build the screening designed experiment test designs and perform statistical analysis [151]. The MATLAB Statistics and Machine Learning Toolbox [152] is utilized for the creation of the optimization experimental test design.

4.2 Neural Network Data Preparation

The IDPS log file data contains 50,000 data observations described by 93 features. Table 18 located in Appendix A, summarizes the number of unique observations, the number of missing observations, and feature data type for each of the 93 features in the IDPS log file dataset. Of the 93 features, 12 are constant valued, and are removed from the dataset without loss of information. After discussions with the agency sponsoring this work, we select the features shown in Table 2 for UANN anomaly detection.

Table 2. ANN Anomaly Detection Features

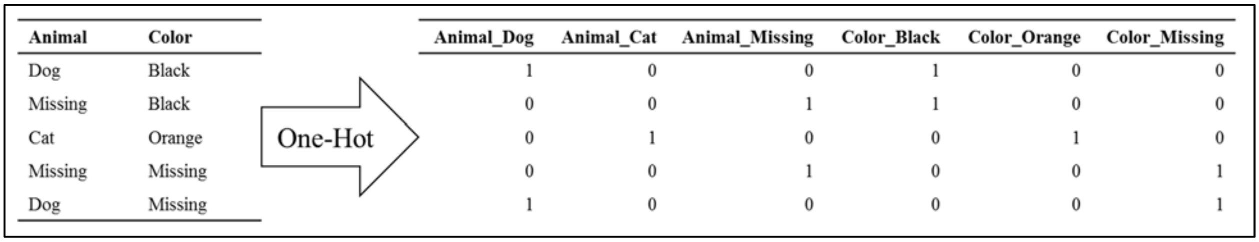
Feature	Class	Number of Missing Observations	Number of Unique Observations
CATEGORYBEHAVIOR	character	205	14
CATEGORYOBJECT	character	222	10
CATEGORYSIGNIFICANCE	character	134	10
CATEGORY_EVENT	character	64	34
COUNTRY_SRC	character	1192	49
EVENTID_DEVICE	character	0	23
EVENTNAME	character	0	47
SEVERITY_AGENT	character	0	3
IP_DST	character	15701	25208
IP_SRC	character	0	9063
PORT_DST	integer	4409	7543
PORT_SRC	integer	22255	4620
PRIORITY_EVENT	integer	0	7
COUNT_EVENT	integer	0	565

The ANN dataset contains 50,000 observations described by the 14 features. To adapt the dataset for use with NNs we first impute the missing observations. For the all character features except *IP_DST*, we replace missing observations with the text string ‘missing’ using [149], [150]. For each of the *IP_DST*, *IP_SRC*, *PORT_DST* and *PORT_SRC* features, we add a column to the dataset with an indicator value of '1' indicating missing data, and a value of '0' for non-missing data. Missing data imputation is completed with [153]. No additional features require missing data imputation.

The *IP_DST*, *IP_SRC* features describe the destination and source IP addresses of the network traffic generating the log file. Within the IDPS log file dataset the 32-bit IP addresses are provided in the human readable notation, for example "127.012.252.001", in which each three-digit number grouping is the decimal representation of an 8-bit number. The IP addresses are converted from the human readable notation into their 32-bit binary value representation using [154]. The 32-bit binary numbers are then split into 32 unique integer valued columns using [153], where each column corresponds to a digit of the binary number. Each unique IP address in the final ANN dataset is represented by 33 columns, one column indicating missing values, and 32 columns representing the 32-bit binary representation of the IP addresses.

The *PORT_SRC* and *PORT_SRC* features describe the computer port numbers of the source and destination network traffic. The port numbers provided are the decimal representation of 16-bit binary numbers. The decimal valued port number are converted to their 16-bit binary representation using [155], then split into 16 unique integer valued columns using [153]. Each port number in the final UANN dataset is represented by 17 integer valued columns, one column indicating missing observations, and 16 columns representing the 16-bit binary representation of the port number.

The remaining character columns are converted into numeric columns using one-hot encoding. One-hot encoding creates an integer valued indicator column for every unique observations within all character features. For each observation within a feature, the corresponding indicator column where the observation value matches the indicator column receives a value of “1” with all other indicator columns a value of “0”. Figure 15 depicts an example of the one-hot encoding process using the notional character features of *Animal* and *Color*.



Animal	Color
Dog	Black
Missing	Black
Cat	Orange
Missing	Missing
Dog	Missing

Animal_Dog	Animal_Cat	Animal_Missing	Color_Black	Color_Orange	Color_Missing
1	0	0	1	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1
1	0	0	0	0	1

Figure 15. One-Hot Encoding

As evident from Figure 15, the dataset after one-hot encoding retains the same number of observations, however the number of features increases. One-hot encoding the remaining character features in the IDPS log file dataset using [156] provides us our final UANN dataset containing 50,000 observations described by 292 numeric features.

We continue NN data preparation by randomly splitting the UANN dataset into three test and training subsets. Each subset contains all 50,000 observations, with 85% of observations randomly assigned to the training set and the 15% remaining to the test set. We utilize multiple sets of training and test data to ensure the final UANN selected performs well agnostic to the random splitting process. Recall, the dataset is unsupervised and is likely to contain anomalous observations. During splitting, there exists the chance that anomalous observations are assigned to the training dataset. Consequently, the UANN may learn to reconstruct the anomalous data, resulting in low outlier factor scores for those observations. Ensuring the UANN generalizes well to all training-testing sets reduces the probability of learning the anomalous observations and maximizes the regularization capability of the NN.

We conclude the UANN data preparation with data scaling. Each training and test subset of data is independently scaled to the intervals discussed in section 3.5.8.9 using [157].

4.3 Hyperparameter Screening Designed Experiments

Screening designed experiments are used to identify the UANN hyperparameters which significantly affect the generalization error. Utilization of screening experiments prior to hyperparameter optimization improves the efficiency of subsequent experiments by only focusing on the hyperparameters with significant effect on the generalization error. Table 3 summarizes the hyperparameters and the levels evaluated in the main effect screening designed experiments.

Table 3. Main Effect Screening Test Design Factors

Hyperparameter	**Type**	**Levels**
Number of Hidden Layers	Integer	1 ,2, 3
Dropout	Categorical	True, False
Input Dropout Rate	Continuous	0.2, 0.8
Hidden Dropout Rate	Continuous	0.2, 0.8
Adaptive Rate	Categorical	True, False
Rho	Continuous	0.9, 0.999
Epsilon	Continuous	1E-10, 1E-6
Activation Function	Categorical	ReLU, Tanh
Neurons Per Hidden Layer	Integer	73, 213
Learning Rate	Continuous	5E-5, 5E-2
Rate Annealing	Continuous	0, 1E-4
Rate Decay	Continuous	0, 2
Momentum Start	Continuous	0.25, 0.75
Momentum Stable	Continuous	0.9, 0.999
Momentum Ramp	Continuous	500, 50000
Nesterov Accelerated Gradient	Categorical	True, False
L1	Continuous	0, 1
L2	Continuous	0, 1
Max W2	Continuous	5, 500
Initial Weight Distribution	Categorical	Uniform, Uniform Adaptive, Normal
Average Activation	Continuous	0.05, 0.5
Sparsity Beta	Continuous	0.5, 2
Minibatch Size	Integer	1, 10000
Shuffle Training Data	Categorical	True, False
Data Scale	Categorical	[0, 1], [-0.5, 0.5], [-1, 1]
Test/Training Set	Categorical	1, 2, 3

Ideally, we would pursue one large designed experiment to screen all the hyperparameters listed in Table 3, however this is not possible due to the existence of dependencies amongst the hyperparameters. When adaptive rate is set to the value of true, then we are required to also specify the rho and epsilon hyperparameters. If the adaptive rate is set to false, the learning rate, rate annealing, rate decay, momentum start, momentum stable, and moment ramp are required. The adaptive rate hyperparameter inherently affects which set of hyperparameters require specification. Similarly, if dropout is set to true, we are required to specify the input dropout rate, and the hidden dropout rate, whereas if dropout is set to false, the dropout rates are not required. When testing more than one hidden layer we are required to specify the number of hidden neurons for each hidden layer, and if dropout is set to true, we are also required to specify a hidden dropout ratio for each hidden layer. Figure 16 outlines the hyperparameter dependencies. In order to perform hyperparameter screening, we first must identify the appropriate design considering the dependencies of the hyperparameters.

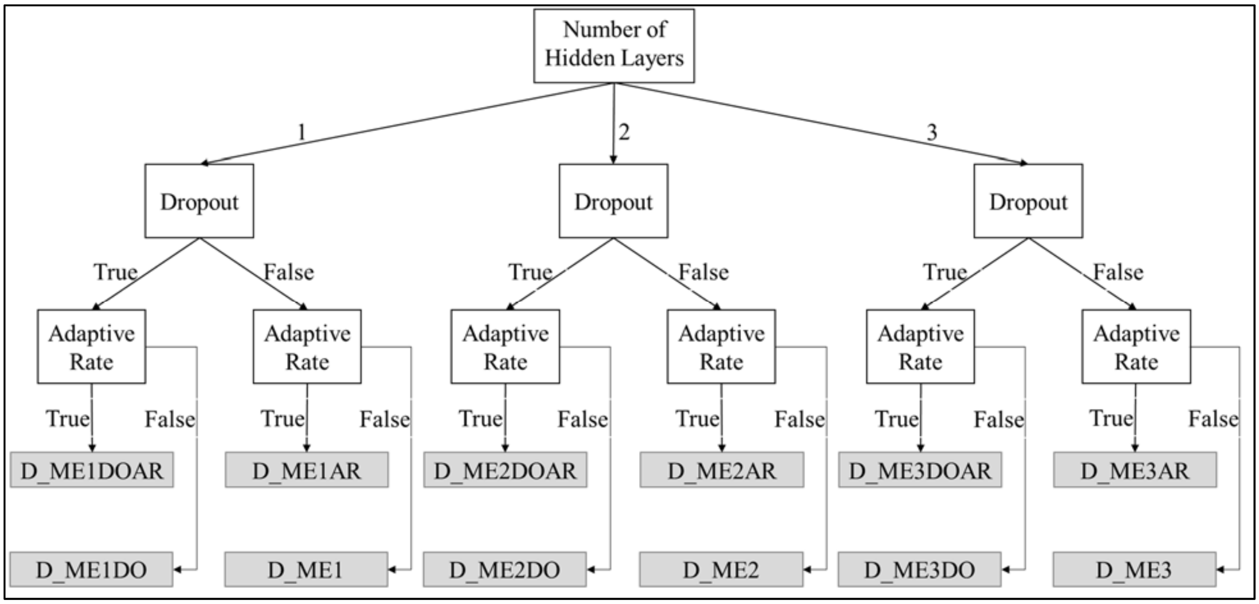


Figure 16. Hyperparameter Dependencies

To account for the hyperparameter dependencies we specify 12 screening designed experiments. Within Figure 16, each grey filled box corresponds to a different screening experimental test design constructed using the hyperparameters and levels in Table 3. The test designs are computer generated D-Optimal main effects only screening designs augmented with computer selected center points, constructed using [151]. Computer generated designs are utilized due to the mixture of continuous, categorical, and integer hyperparameters, as well as an inconsistent number of levels tested for each hyperparameter. D-Optimal designs are utilized to minimize the size of the confidence interval around each hyperparameter estimate. Each screening experiment is designed to detect hyperparameter significance ($\alpha = 0.05$ level) with 80% probability as shown in the power column of Table 4 which displays basic test design information for each test design. We augment the main effects only design with additional center points to achieve the desired hyperparameter power. The number of required center points and the total number of required test points varies due to the changing number of hyperparameters evaluated across the test designs. The D-Efficiency metric measures the relative D-optimality of each test design compared to an ideal design. The 12 hyperparameter dependency test designs are provided in Appendices B-M.

Table 4. D-Optimal Main Effect Screening Design Metrics (α=0.05)

Test Design	**Test Points**	**Minimum Hyperparameter Power**	**D-Efficiency**
D_ME1DOAR	30	88.3%	87.6%
D_ME1DO	36	95.3%	88.6%
D_ME1AR	30	90.1%	88.6%
D_ME1	32	88.5%	86.6%
D_ME2DOAR	30	84.8%	86.3%
D_ME2DO	36	93.6%	86.9%
D_ME2AR	30	90.8%	88.8%
D_ME2	36	95.2%	88.7%
D_ME3DOAR	37	95.6%	86.4%
D_ME3DO	37	92.4%	84.4%
D_ME3AR	31	89.5%	86.6%
D_ME3	37	95.4%	86.7%

After test design execution we utilize JMP to select the test design architecture with the smallest generalization error for selection of the most promising mutually exclusive UANN architecture for subsequent hyperparameter screening. The D_ME1DOAR architecture is identified as having the best generalization error (see section 5.2).

To screen the categorical hyperparameters of the D_ME1DOAR design, we augment the design with additional test points to detect 2-way interactions between hyperparameters and add center points to the test design to detect possible curvature in the numeric hyperparameters. To detect 2-way interactions amongst hyperparameter we use JMP's design augmentation using D-efficiency as the design generation criteria. Design augmentation for 2-way interaction adds 218 test points to the D_ME1DOAR test design. In order to more robustly capture potential curvature for continuous hyperparameters we construct a full factorial designed experiment for the D_ME1DOAR categorical features in which the continuous features are all set to the center point value. Table 5 outlines the hyperparameters and levels for the full factorial center point design. Hyperparameters marked with the superscript "+" are those factors which are fixed as a result of the selection of the D_ME1DOAR test design for augmentation.

Table 5. Center Point Full Factorial Design Factors

Hyperparameter	**Type**	**Levels**
Number Hidden Layers$^+$	Integer	1
Dropout$^+$	Categorical	True
Input Dropout Rate	Continuous	0.5
Hidden Dropout Rate	Continuous	0.5
Adaptive Rate$^+$	Categorical	True
Rho	Continuous	0.99
Epsilon	Continuous	1.00E-08
Activation Function	Categorical	ReLU, Tanh
Neurons Per Hidden Layer	Integer	143
Nesterov Accelerated Gradient	Categorical	True, False
L1	Continuous	0.5
L2	Continuous	0.5
Max W2	Continuous	252.2
Initial Weight Distribution	Categorical	Uniform, Uniform Adaptive, Normal
Average Activation	Continuous	0.275
Sparsity Beta	Continuous	1.25
Minibatch Size	Integer	5000
Shuffle Training Data	Categorical	True, False
Data Scale	Categorical	[0, 1], [-1, 1]
Test/Training Set	Categorical	1, 2, 3

The center point full factorial design requires an additional 144 test points. The 2-way interaction augmentation and the center point augmentation designs are added to the D_ME1DOAR design resulting in the addition of 362 additional test points to the 30 previously completed in during D_ME1DOAR testing. The final hyperparameter screening design has 392 test points, and can detect curvature as well as hyperparameter and all 2-way hyperparameter interaction effect significance ($\alpha = 0.05$ level) with 99.9% probability. The full screening design is provided in Appendix N.

We analyze the hyperparameter screening design using ANOVA at the $\alpha = 0.05$ level of significance. Our goal is to determine what levels of the categorical hyperparameters minimize generalization error. In subsequent hyperparameter optimization we consider the numeric hyperparameters, and only two categorical hyperparameters, the activation function, and the test/training set. For the other categorical features, we set the hyperparameter values at the screening design identified optimum levels.

4.4 Hyperparameter Optimization Designed Experiments

The goal of hyperparameter optimization is to determine the numeric hyperparameter values resulting in the minimum generalization error. We use a conventional response surface design, the orthogonal central composite design (CCD) created using [152], to optimize the numeric hyperparameters. The CCD design selected for the numeric hyperparameter optimization is based around a $2^{11-4}_{\text{Res V+}}$ fractional factorial requiring 128 factorial test points. To maintain orthogonality the design is augmented with 22 axial points and 28 center points, bringing the total test points for numeric hyperparameter optimization to 178. Table 6 lays out the numeric hyperparameter levels tested in the CCD.

Table 6. Numeric CCD Hyperparameter Factors and Levels

	Levels				
	Lower Axial	Lower Factorial	Center	Upper Factorial	Upper Axial
Hyperparameter	**-3.3635**	**-1**	**0**	**1**	**3.3635**
Neurons Per Hidden Layer	10	32	42	52	73
Rho	0.800	0.870	0.900	0.930	1.000
Epsilon	1.00E-15	2.23E-13	3.16E-13	4.10E-13	1.00E-10
Input Dropout Rate	0.010	0.768	0.105	0.133	0.200
Hidden Dropout Rate	0.010	0.768	0.105	0.133	0.200
L1	0.00E+00	2.97E-06	1.00E-05	1.70E-05	1.00E-07
L2	0.00E+00	2.97E-06	1.00E-05	1.70E-05	1.00E-07
Max W2	1	2.405	3.000	3.595	5.000
Average Activation	0.005	0.021	0.028	0.034	0.050
Sparsity Beta	0.05	0.208	0.275	0.342	0.5
Minibatch Size	1	3513	5000	6487	10000

The 178 test point CCD numeric hyperparameter optimization experiment is replicated for each activation function ($ReLU$ and Tanh) level as well as replicated for each of the test/training sets (1, 2, 3). Table 7 depicts the categorical hyperparameter levels tested in the CCD optimization designed experiment.

Table 7. Categorical CCD Hyperparameter Factors and Levels

Hyperparameter	**Levels**
Activation Function	ReLU, Tanh
Test/Training Set	1, 2, 3
Nesterov Accelerated Gradient	True
Initial Weight Distribution	Uniform
Shuffle Training Data	True
Data Scale	[0, 1]

The CCD hyperparameter optimization test design contains 1068 total test points and is provided in Appendix O.

After executing the CCD designed experiment we use ANOVA to construct two statistical models of the generalization error as a function of the hyperparameters. The first model is the full model containing all hyperparameters (except test/train sets), all 2-way hyperparameter interactions, and all numeric hyperparameter quadratic effects. The test/training set is not considered as a hyperparameter to ensure the final statistical model fits well regardless of the test/training set considered. The second model is a parsimonious model containing only significant hyperparameters, interactions, and quadratic effects ($\alpha = 0.05$ level). For both the full and parsimonious statistical models, we use JMP to identify the optimum hyperparameter values resulting in the lowest generalization error. We then use the statistical model predicted optimum hyperparameter values to construct UANNs and validate the statistical model predicted generalization error to the actual generalization error obtained using UANNs when tested against all three test/training sets. If the generalization error of the statistical model is consistent with the generalization error of the UANNs, and the generalization errors are consistent for each of the test/training sets we conclude hyperparameter optimization. If previous conditions do not hold, we use the information gained from the statistical models to modify the hyperparameter levels and execute a follow-on phase of hyperparameter optimization testing using a new 1068 test point CCD. We continue to iterate through CCD experimental designs until a suitable set of hyperparameter values are found resulting in consistent generalization error between test/training sets and similar ANOVA model generalization error prediction with actual UANN generalization errors.

4.5 Graphical Outlier Detection

After finding the optimum hyperparameter values, we proceed with anomaly detection. We first scale the entire UANN dataset into the optimum interval found during the hyperparameter screening designed experiment. The dataset used for anomaly detection is not split into training and test subsets, instead, to evaluate the UANN reconstruction performance we use a random dataset. For each of the 292 feature within the UANN dataset, we randomly and independently select one of the 50,000 data values. After sampling a data value from each feature, we combine the randomly drawn data points into a random observation. We repeat this process until we obtain the random dataset composed of 50,000 observations. This random dataset is then scaled into the same interval as the UANN dataset.

ANN with a bottleneck layer are forced to reconstruct the most salient features of the dataset from a compressed internal representation. Random data should have no salient features when reconstructed by an UANN. Comparing the distribution of outlier factor scores points from the real dataset and a random dataset enables us to validate that the UANN is effectively learning the hidden structures within the real data.

We use the full scaled UANN dataset to train the UANN with the optimum hyperparameter values obtained in the screening and optimization experimental designs. Once the UANN is trained, we calculate the outlier factor score of each observation within the UANN dataset and the random dataset. The outlier factor scores for each UANN dataset observation and the random dataset are plotted on a histogram for visual identification of anomalous observations. The UANN dataset outlier factor scores closest to the outlier factor score of the random data are declared as anomalous observations.

V. Results

5.1 Chapter Overview

In this chapter we summarize notable results from the hyperparameter dependency testing, hyperparameter screening ANOVA, and hyperparameter optimization ANOVA. We use a confidence level of 0.05 for all statistical evaluations. The identified optimum hyperparameter values are used to construct the UANN for anomaly detection. We present the histogram of the UANN dataset outlier factor scores and the random data factor scores for anomaly detection. The 10 observations with the largest outlier factor scores are displayed.

The NNs in this work are constructed using the h2o.ai software [158] within the RStudio IDE [150], using the R programming language [149]. All UANNs are trained using the MSE objective function (Equation (21)). UANNs are trained until we meet one of the following training termination criteria: training time exceeds 1-hour (3,600 seconds), 100 training epochs, the MSE of the training set decreases below $1 * 10^{-8}$, or the training MSE fails to change by at least $1 * 10^{-8}$ for five consecutive epochs.

5.2 Hyperparameter Screening Results

A summary of the results of the hyperparameter dependency testing is displayed in Table 8. During UANN training we discovered using some hyperparameter values resulted in exponential growth of the weight and bias with the network. This led to unstable NN architectures where training was prematurely terminated. Generally we observe when the adaptive rate hyperparameter is set to the true level, (test designs containing 'AR' within their name) the stability of the UANN improves. This may be due to inappropriate hyperparameter values for the momentum and learning rate, however, more testing is required to substantiate this claim.

The average test dataset MSE (generalization error) is presented along with the standard deviation for those UANNs which converged (left of dashed line). The D_ME1DOAR and D_ME3DOAR designs show the lowest average generalization error and the lowest standard deviations. Also notable is the poor average generalization error and large standard deviations of the D_ME1AR and D_ME3AR designs. The poor performance of these designs is caused by a few outlier points. The presence of the dropout (test designs containing 'DO' within their name), appears to improve regularization as all the test designs with outliers do not utilize dropout.

After removing the outliers we obtain the average generalization errors and standard deviation show right of the dashed line in Table 8. We observe average generalization errors and standard deviations of the same magnitude across the 12 test designs.

Table 8. Hyperparameter Dependency UANN Results Summary

Test Design	**Test Points**	**Unstable NNs**	**Average Test MSE**	**Standard Deviation Test MSE**	**Outliers Removed**	**Average Test MSE**	**Standard Deviation Test MSE**
D_ME1DOAR	30	3	0.320	0.246	0	0.320	0.246
D_ME1DO	36	11	0.389	0.265	0	0.389	0.265
D_ME1AR	30	2	49.889	204.319	3	0.399	0.364
D_ME1	32	11	0.530	0.421	0	0.530	0.421
D_ME2DOAR	30	0	0.380	0.278	0	0.380	0.278
D_ME2DO	36	14	0.684	0.760	0	0.684	0.760
D_ME2AR	30	0	2.492	11.149	1	0.458	0.401
D_ME2	36	10	1.168	2.049	1	0.790	0.713
D_ME3DOAR	37	1	0.322	0.238	0	0.322	0.238
D_ME3DO	37	17	0.463	0.432	0	0.463	0.432
D_ME3AR	31	1	116.357	634.418	1	0.529	0.615
D_ME3	37	16	1.242	2.937	1	0.606	0.390

Figure 17 depicts a plot of the generalization error for each of the 12 hyperparameter dependency test designs after outlier removal. There is insufficient evidence to claim the D_ME1DOAR test design has the smallest average generalization error statistically.

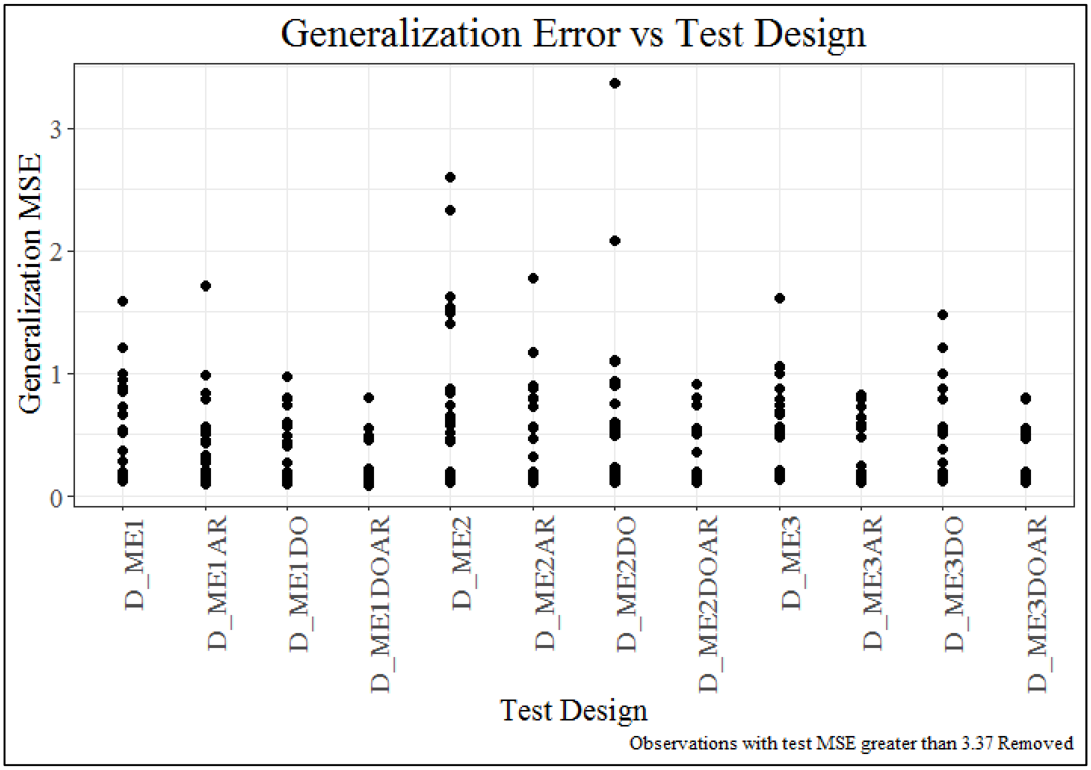


Figure 17. Generalization Error of Hyperparameter Dependency UANNs

Of the smallest 20 generalization errors observed, five (including the smallest observed generalization error) are found to be from UANNs within the D_ME1DOAR test design, as shown in Figure 18.

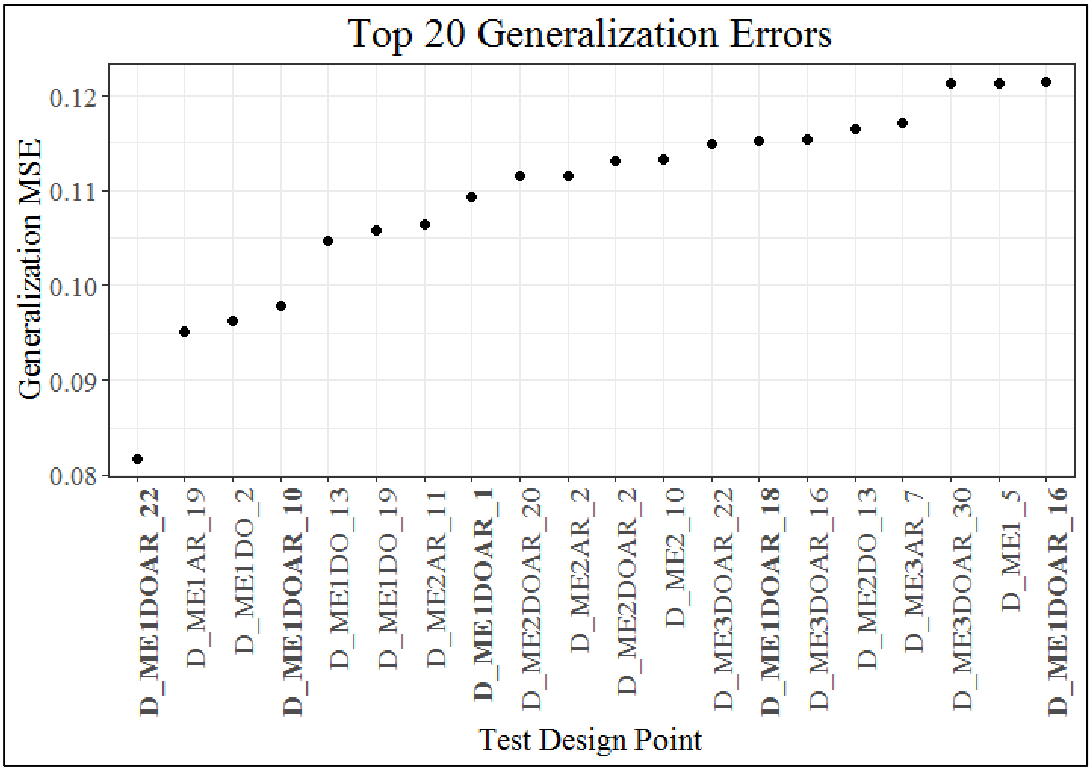


Figure 18. Top 20 Hyperparameter Dependency UANNs Generalization Errors

Analysis in JMP indicates the D_ME1DOAR hyperparameter dependency design contains the most desirable generalization errors, therefore, the D_ME1DOAR design is selected for hyperparameter screening.

Prior to conducting a more comprehensive screening experiment, we analyzed the D_ME1DOAR experiment using ANOVA. Analysis revealed the most signifigant hyperparameter to the generalization error was the data scale. Scaling the data to the [-1, 1] interval produced signficantly worse generalization error. For subsequent testing we removed this level from the data scale hyperparameter, and augmented the D_ME1DOAR design as discussed in section 4.3.

Analysis of the D_ME1DOAR screening designed experiment used forward stepwise regression in which only model terms with a significance less than the $\alpha = 0.05$ level are included. The initial models evaluated contained significant outliers, which were removed. After outlier removal the forward stepwise regression was recompleted. The outlier removal and stepwise regression was repeated until a suitable model was found. The derived statistical model required the removal of 10 outliers. This model contained all categorical hyperparameters except for shuffle training data and all numeric hyperparameters except for the L2 hyperparameter. These model terms were added to the model, despite their lack of significance, to obtain the final model for the screening experiment design. Non-significant model terms are retained to identify the hyperparameter levels with the lowest generalization error for all evaluated hyperparameters. We analyzed the statistical model to determine the hyperparameter values resulting in the smallest estimated generalization error. The identified values are displayed in Table 9.

Table 9. Optimum Screening Design Hyperparameter Levels

Hyperparameter	**Level**
Input Dropout Rate	0.2
Hidden Dropout Rate	0.2
Rho	0.9
Epsilon	1.00E-10
Activation Function	Tanh
Neurons Per Hidden Layer	73
Nesterov Accelerated Gradient	True
L1	0
L2	0
Max W2	5
Initial Weight Distribution	Uniform
Average Activation	0.05
Sparsity Beta	0.5
Minibatch Size	10000
Shuffle Training Data	True
Data Scale	[0, 1]
Test/Training Set	2

We used the numeric hyperparameter values from Table 9 to generate the levels for numeric hyperparameter optimization (section 4.4). The categorical hyperparameters from the table are set to their optimal screening values for numeric optimization, except for the activation function and the test/training set which are retained as test design factors for additional evaluation.

5.3 Hyperparameter Optimization Results

Leveraging the information gained in the hyperparameter screening designed experiment we developed a CCD to optimize the numeric hyperparameters. The hyperparameter levels tested are provided in Table 6 and Table 7. We analyzed the CCD design in two ways. The first analysis method used forward stepwise regression in which only model terms with a signifigance less than the $\alpha = 0.05$ level are included. This reduced model with only significant terms is called the parsimonious model. The second analysis method, called the full model, used all hyperparameters, 2-way hyperparameter interactions and numeric hyperparameter quadratic effects. For both the full and parsimonious models, we removed 23 outliers prior to finalizing the statistical models. For both statistical models we determined the optimum hyperparameter values resulting in the smallest generalization error, the values are provided in Table 10.

Table 10. CCD Optimum Hyperparameter Values

Hyperparameter	**Full**	**Parsimonious**
Activation Function	Tanh	Rectifier
Neurons Per Hidden Layer	73	73
Rho	0.8	0.999999
Epsilon	1.00E-10	5.01E-11
Input Dropout Rate	0.2	0.105$^{\wedge}$
Hidden Dropout Rate	0.01	0.010
L1	0.000017	1.00E-5$^{\wedge}$
L2	0	1.00E-5$^{\wedge}$
Max W2	1	1
Average Activation	0.005	0.0275$^{\wedge}$
Sparsity Beta	0.5	0.275$^{\wedge}$
Minibatch Size	10000	5000$^{\wedge}$

In the parsimonious model hyperparameter models marked with a superscript “^” are those hyperparameters not significant, however are included at their center point values to specify all required hyperparameters in the UANN. The statistical models estimated generalization error and the corresponding 95% confidence intervals are provided in Table 11.

Table 11. CCD Hyperparameter Statistical Model Generalization Error Prediction

Predicted Mean	**Full**	**Parsimonious**
Test MSE	-4.4230	-0.3791
Lower CI	-7.6661	-0.4636
Upper CI	-1.1799	-0.2946

The statistical models for both the parsimonious and full analyses predict a negative generalization error with a negative upper confidence bound. Any UANN using the hyperparameters in Table 10, are guaranteed to result in generalization error greater than or equal to zero, therefore we conclude these statistical models are not suitable for hyperparameter optimization and validation testing is not required. Leveraging the information gained from the first phase of CCD testing, we modified the hyperparameter levels to the values shown in Table 12 and conducted a second phase of numeric hyperparameter optimization. The phase II CCD test design is provided in Appendix P.

Table 12. Phase II Numeric CCD Hyperparameter Factors and Levels

			Levels		
	Lower Axial	Lower Axial	Lower Axial	Lower Axial	Lower Axial
Hyperparameter	**-3.3635**	**-1**	**0**	**1**	**3.3635**
Neurons Per Hidden Layer	39	73	83	93	127
Rho	0.723	0.760	0.780	0.800	0.847
Epsilon	7.23E-11	8.47E-11	1.20E-10	1.71E-10	4.05E-10
Input Dropout Rate	0.002	0.008	0.009	0.100	0.340
Hidden Dropout Rate	0.002	0.008	0.009	0.100	0.340
L1	4.33E-09	1.00E-06	1.00E-05	1.00E-04	2.31E-02
L2	4.33E-09	1.00E-06	1.00E-05	1.00E-04	2.31E-02
Max W2	0.005	0.021	0.000	3.595	5.000
Average Activation	0.005	0.021	0.028	0.034	0.050
Sparsity Beta	0.05	0.208	0.275	0.342	0.5
Minibatch Size	1	3513	5000	6487	10000

As in the first phase of CCD we derive a parsimonious and full statistical model and identify the hyperparameter values resulting in the minimum generalization error, shown in Table 13. Phase II CCD testing did not result in any outliers for any of the evaluated UANNs.

Table 13. Phase II CCD Optimum Hyperparameter Values

Hyperparameter	**Full**	**Parsimonious**
Activation Function	Rectifier	Rectifier
Neurons Per Hidden Layer	39	127
Rho	0.7128	0.78$^{\wedge}$
Epsilon	7.23E-11	7.23E-11
Input Dropout Rate	0.0024	0.009$^{\wedge}$
Hidden Dropout Rate	0.0336	0.009$^{\wedge}$
L1	0.0231	2.31E-02
L2	0.0231	0.0001$^{\wedge}$
Max W2	0.127	0.9166
Average Activation	0.05	0.0342$^{\wedge}$
Sparsity Beta	0.5	0.05
Minibatch Size	10000	5000$^{\wedge}$

Parsimonious model hyperparameter models marked with a superscript "^" are those hyperparameters not significant, however are included at their center point values to specify all required hyperparameters in the UANN. The phase II CCD statistical models estimated generalization error and the corresponding 95% confidence intervals are provided in Table 14.

Table 14. Phase II CCD Hyperparameter Statistical Model Generalization Error Predictions

Predicted Mean	**Full**	**Parsimonious**
Test MSE	-2.5171	0.0607
Lower CI	-79.3716	0.0540
Upper CI	74.3374	0.0674

The full statistical model predicted generalization error predicts a negative mean value, however the confidence interval around the mean contains positive values. This is most likely due to the size of the interval. The parsimonious model has a positive predicted mean generalization error, and the corresponding confidence interval does not contain zero. There is sufficient evidence to validate the statistical models performance using UANNs with the optimum numeric hyperparameters in Table 13.

We trained 12 validation UANNs using the Table 13 hyperparameters, six UANNs using the optimum hyperparameters for the full model and six using the hyperparameters for the parsimonious model. For each of the three test/training subsets, we construct two UANNs for validation. The validation UANN generalization errors are displayed in Table 15.

Table 15. Phase II CCD Hyperparameter UANN Validation Test Results

Test/Train Set	**Full Model Test MSE**	**Parsimonious Model Test MSE**
1	0.1101	0.0568
2	0.1024	0.0540
3	0.1075	0.0573
1	0.1102	0.0566
2	0.1021	0.0541
3	0.1074	0.0574

Evaluation of the UANN generalization errors shown in Table 15 show consistent performance across the three train/test sets of data. The UANN's trained with the parsimonious statistical model hyperparameters outperform the full statistical model hyperparameters for each of the sets of data. The phase II CCD parsimonious model optimum hyperparameter values align closely with the validated performance of UANNs across the three test/training sets of data. Comparing the confidence intervals of the full model $[0.1028, 0.1104]$ to the interval for the parsimonious model $[0.0544, 0.0577]$, we observe the parsimonious model has a statistically significantly lower generalization error. We terminated our hyperparameter optimization phase and used the parsimonious optimum hyperparameters to train the UANN for anomaly detection.

5.4 Final Anomaly Detection UANN

The anomaly detection UANN is trained using the MSE objective function (Equation (21)). For anomaly detection we modify the training termination criteria to the following: training time exceeds 10-hours (36,000 seconds), 10,000 training epochs, the MSE of the training set decreases below $1 * 10^{-12}$, or the training MSE fails to change by at least $1 * 10^{-12}$ for five consecutive epochs. The hyperparameter levels used for anomaly detection are presented in Table 16.

Table 16. UANN Anomaly Detection Hyperparameter Values

Hyperparameter	Value
Number Hidden Layers	1
Dropout	True
Input Dropout Rate	0.009
Hidden Dropout Rate	0.009
Adaptive Rate	True
Rho	0.78
Epsilon	7.23E-11
Activation Function	Rectifier
Neurons Per Hidden Layer	127
Nesterov Accelerated Gradient	True
L1	0.0231
L2	1.00E-04
Max W2	0.9166
Initial Weight Distribution	Uniform Adaptive
Average Activation	0.3412
Sparsity Beta	0.05
Minibatch Size	5000
Shuffle Training Data	True
Data Scale	[0, 1]

After training we used the UANN to compute the outlier factor score (reconstruction MSE) of each observation of the ANN dataset, and the 50,000 random data observations. The resulting outlier factor scores are depicted in Figure 19.

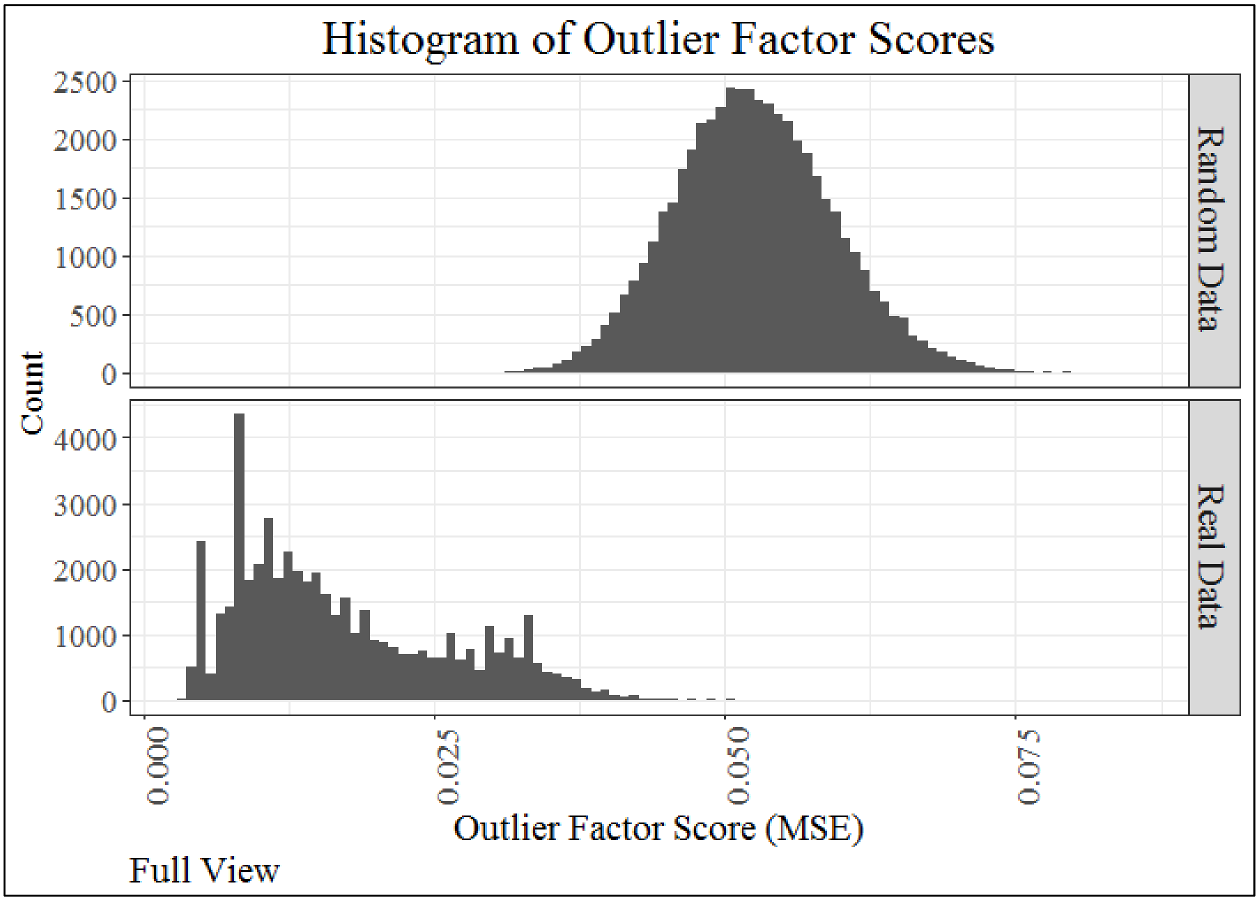


Figure 19. Histogram of Outlier Factor Scores

Examination of the outlier factor scores reveals a distinct separation between the random data and a majority of the actual data, indicating that the UANN is learning the prominent patterns with the ANN dataset. The random data, which should contain no patterns, is reconstructed poorly in comparison to the real data, resulting in a normal distribution appearance. The outlier factor scores in the right-tail of the real data distribution overlap with the random data outlier factor score distribution. The real data observations in the overlap region are data points which are reconstructed by the ANN no better than random data. These observations are those most likely to be anomalies. We present the top ten observations with the greatest outlier factor scores in Table 17.

Table 17. Top 10 Outlier Factor Scores

Outlier Rank	Observation Number	Outlier Factor Score
1	2182	0.059870
2	6391	0.055390
3	23235	0.053769
4	8012	0.052579
5	10759	0.052290
6	8098	0.051338
7	8414	0.051317
8	378	0.051035
9	14908	0.050599
10	2711	0.050502

VI. Conclusions and Recommendations

6.1 Conclusion and Contributions

In this work we present a NN based approach to anomaly detection designed to augment current detection methods employed by the sponsoring agency, enhancing their capability to detect malicious network activity prior successfully adversary misconduct. Using an UANN enables the sponsor to identify specific anomalous observations for further analysis by computer security experts. A DOE testing process provides a simple method for identifying and selecting the optimal UANN hyperparameters for anomaly detection. The use of multiple test and training sets of data enhances the regularization capability of the UANN. Finally, the graphical depiction of the real and random data validates the UANN is effectively learning the hidden structures within the real data.

6.2 Future Research

We recommend the following topics for future research. In this work we employed a sequential DOE process, first screening and selecting the significant categorical hyperparameters using D-Optimal computer generated designs, and then optimizing the numeric hyperparameters using CCDs. The goal of sequential testing being the efficient identification of the optimum hyperparameter values. Future work should examine the utility and efficiency of using computer generated space filling designs for optimal hyperparameter identification.

The IDPS log file dataset is primarily composed of categorical features, which require encoding to a numeric representation prior to NN analysis. Using one-hot encoding increases the number of features of the NN dataset and reduces the utility of current NN methods for determining feature importance. It is recommended that future research identify a more efficient categorical feature encoding methodology to reduce the number of features required in the NN dataset. Additionally, future work should identify a method for determining feature importance agnostic to the categorical feature encoding system employed.

The ANN explored in this work all contained an under-complete bottleneck layer in which we restricted the number of neurons to be smaller than the number of input features. Alternate ANNs exist in which the number of neurons in the hidden layers are not restricted. Instead, stronger regularization and sparsity hyperparameters are utilized to restrict the capability of NN to perfectly recreate the input data at the output layer. Additional research should examine the capability of sparse, over-complete ANNs to perform anomaly detection.

Plotting the outlier factor scores of the real IDPS data along with randomly generated data provides a simple, intuitive method for the identification of anomalous observations within the dataset. Real data observations with outlier factor scores of the same magnitude as random data scores are reconstructed poorly by the anomaly detection autoencoder and are considered anomalous observations. An outlier factor score for each real data observation is obtained. Future research should identify a robust method for automatically determining an appropriate cutoff outlier factor score value such that scores lower than the cutoff value are considered normal, and score greater than the cutoff are considered anomalous.

Finally, we utilized the h2o.ai software for construction of all NNs. While an excellent choice for novice NN practitioners, the h2o.ai software is limited for ANN implementations. Prior to real-world implementation, additional research should examine software alternatives such as, Tensorflow, Keras, MXNet, etc.

Appendix A: IDPS Features Summary

Table 18. IDPS Log File Data Summary

Feature	Class	Number of Missing Observations	Number of Unique Observations
Id	character	0	49984
Timestamp	numeric	0	16
Data Type	character	0	1
Visibility	character	0	1
ACTIONTAKEN_DEVICE	character	49997	2
ARCSIGHTAGENTTYPE	character	0	1
ASN_DST	character	21303	59
ASN_SRC	character	9395	284
ASSETID_CUSTOMER	character	45701	2
AUDITTRAIL_EVENTANNOTATION	character	0	3
CATEGORYBEHAVIOR	character	205	14
CATEGORYGROUP	character	27749	3
CATEGORYOBJECT	character	222	10
CATEGORYOUTCOME	character	213	4
CATEGORYSIGNIFICANCE	character	134	10
CATEGORYTECHNIQUE	character	48829	9
CATEGORY_EVENT	character	64	34
COCOM_DST	character	18170	6
COCOM_SRC	character	2634	7
COUNTRYCODE_DST	character	18255	7
COUNTRYCODE_SRC	character	7638	42
COUNTRY_DST	character	18165	11
COUNTRY_SRC	character	1192	49
COUNT_EVENT	integer	0	565
EVENTID_DEVICE	character	0	23
EVENTNAME	character	0	47
EXTERNALID_CUSTOMER	character	45701	2
FILENAME	character	530	41
FILEPATH	character	530	40
FLAGS_EVENTANNOTATION	character	49998	2
GEOCODE_DST	character	18383	22
GEOCODE_SRC	character	14588	84
GEOLOCATIONNAME_DST	character	18383	67
GEOLOCATIONNAME_SRC	character	14589	264
HOSTNAME_AGENT_FQDN	character	0	1
HOSTNAME_AGENT_FQDN_REVERSE	character	0	1
HOSTNAME_DEVICE_FQDN	character	64	6
HOSTNAME_DEVICE_FQDN_REVERSE	character	64	6

Feature	Class	Number of Missing Observations	Number of Unique Observations
HOSTNAME_DST	character	47468	328
HOSTNAME_DST_REVERSE	character	47468	328
HOSTNAME_SRC	character	47722	144
HOSTNAME_SRC_REVERSE	character	47722	144
IPBRANCHCATEGORY_DST	character	18165	3
IPBRANCHCATEGORY_SRC	character	1192	3
IP_AGENT	character	0	1
IP_AGENT_ORIGINAL	character	0	2
IP_DEVICE	character	0	1
IP_DST	character	15701	25208
IP_SRC	character	0	9063
LABEL_STR_CUSTOM1_HBSSALERTS	character	33240	6
LABEL_STR_CUSTOM2_HBSSALERTS	character	115	3
LABEL_STR_CUSTOM3_HBSSALERTS	character	42494	4
LABEL_STR_CUSTOM4_HBSSALERTS	character	42496	3
LABEL_TIME_CUSTOM1_HBSSALERTS	character	49948	3
LATITUDE_DST	numeric	18346	81
LATITUDE_SRC	numeric	1192	405
LOCALITY	character	0	1
LONGITUDE_DST	numeric	18346	82
LONGITUDE_SRC	numeric	1192	404
ORGANIZATION_OWNER_DST	character	22569	44
ORGANIZATION_OWNER_SRC	character	32557	99
PORT_DST	integer	4409	7543
PORT_SRC	integer	22255	4620
POSTALCODE_DST	integer	18383	81
POSTALCODE_SRC	character	25117	236
PRIORITY_ASSET	integer	0	1
PRIORITY_EVENT	integer	0	7
PRODUCTNAME	character	0	1
SEVERITY_AGENT	character	0	3
SEVERITY_DEVICE	character	0	1
STR_CUSTOM1_HBSSALERTS	character	33267	23
STR_CUSTOM2_HBSSALERTS	character	115	36
STR_CUSTOM3_HBSSALERTS	character	42494	3211
STR_CUSTOM4_HBSSALERTS	character	42493	5
TIME_CUSTOM1_HBSSALERTS	POSIXt	49950	47
TIME_END	POSIXt	0	26467
TIME_END_EVENTANNOTATION	POSIXt	0	26467
TIME_MODIFIED_EVENTANNOTATION	POSIXt	0	19724
TIME_RECEIPT_EVENTANNOTATION	POSIXt	0	19696

Feature	Class	Number of Missing Observations	Number of Unique Observations
TIME_STAGEUPDATE_EVENTANNOTATION	POSIXt	0	19724
TIME_START	POSIXt	0	26508
TYPE_EVENTSOURCE	character	0	2
URI_CUSTOMER	character	43144	4
URI_DEVICE	character	717	2
URI_DST	character	16016	76
URI_SRC	character	3871	207
USERNAME_DST	character	49944	6
USERNAME_SRC	character	49989	2
VENDORNAME_DEVICE	character	0	1
VERSION_AGENT	character	64	4
VERSION_DEVICE	character	64	3
VERSION_EVENTANNOTATION	integer	0	2
ZONENAME_AGENT	character	64	3

Appendix B: D_ME1DOAR

Table 19. D_ME1DOAR Test Matrix

Activation Function	Neurons Per Hidden Layer	Rho	Epsilon	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
RectifierWithDropout	73	0.9	1.00E-10	TRUE	0.2	0.8	0	1	5	Uniform	0.5	2	1	TRUE	[0, 1]	2	0.1093	0.0566	50.1	
RectifierWithDropout	219	0.999	1.00E-10	FALSE	0.2	0.2	0	0	500	Normal	0.5	0.5	1	FALSE	[-0.5, 0.5]	2	0.1990	0.2470	128.9	
TanhWithDropout	219	0.999	1.00E-10	TRUE	0.8	0.2	1	0	5	Uniform	0.5	2	10000	TRUE	[-1, 1]	2	0.5553	0.5505	189.2	
RectifierWithDropout	73	0.9	1.00E-10	FALSE	0.2	0.2	1	0	5	Uniform Adaptive	0.05	2	10000	TRUE	[-0.5, 0.5]	3	0.1968	0.2448	114.4	
TanhWithDropout	219	0.9	1.00E-10	TRUE	0.2	0.8	0	1	5	Normal	0.5	0.5	1	FALSE	[-1, 1]	3	0.4964	0.2873	124.3	
TanhWithDropout	219	0.999	1.00E-06	FALSE	0.2	0.2	1	1	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	1	0.1927	0.1465	263.5	
RectifierWithDropout	219	0.9	1.00E-10	FALSE	0.2	0.2	0	1	500	Normal	0.05	2	10000	TRUE	[-1, 1]	1	0.7995	0.9707	89.1	
TanhWithDropout	73	0.999	1.00E-10	FALSE	0.8	0.2	1	1	5	Normal	0.05	0.5	10000	FALSE	[-1, 1]	2	0.5584	0.5614	100.6	
TanhWithDropout	219	0.999	1.00E-10	TRUE	0.2	0.8	1	0	500	Uniform Adaptive	0.5	2	10000	FALSE	[-0.5, 0.5]	2	0.1990	0.2469	155.9	
TanhWithDropout	219	0.999	1.00E-10	FALSE	0.2	0.8	0	0	5	Uniform	0.05	0.5	1	TRUE	[0, 1]	1	0.0979	0.0436	117.0	
RectifierWithDropout	73	0.999	1.00E-06	TRUE	0.2	0.8	1	1	500	Normal	0.05	2	1	TRUE	[-1, 1]	2	0.7960	0.9878	117.1	
TanhWithDropout	73	0.9	1.00E-06	TRUE	0.2	0.2	0	0	500	Uniform	0.5	2	10000	FALSE	[-1, 1]	1	0.4586	0.2954	60.0	
RectifierWithDropout	219	0.999	1.00E-06	TRUE	0.8	0.8	0	0	5	Normal	0.05	2	10000	FALSE	[0, 1]	1				Exp Growth
RectifierWithDropout	73	0.9	1.00E-06	TRUE	0.8	0.2	1	0	5	Normal	0.5	0.5	1	TRUE	[-0.5, 0.5]	1	0.1999	0.2427	115.9	
RectifierWithDropout	219	0.999	1.00E-06	TRUE	0.8	0.2	0	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	3	0.2230	0.1950	56.8	
TanhWithDropout	73	0.999	1.00E-10	TRUE	0.8	0.8	0	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 0.5]	1	0.1215	0.0748	30.9	
RectifierWithDropout	73	0.999	1.00E-10	FALSE	0.8	0.8	0	0	5	Uniform Adaptive	0.5	2	1	FALSE	[-1, 1]	3				Exp Growth
TanhWithDropout	219	0.9	1.00E-06	FALSE	0.8	0.8	0	1	5	Normal	0.5	2	10000	TRUE	[-0.5, 0.5]	2	0.1152	0.0711	136.2	
RectifierWithDropout	219	0.999	1.00E-06	TRUE	0.2	0.8	1	1	5	Uniform	0.05	0.5	10000	FALSE	[-0.5, 0.5]	3				Exp Growth
RectifierWithDropout	219	0.9	1.00E-06	FALSE	0.8	0.8	1	0	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-1, 1]	2	0.7960	0.9878	53.9	
TanhWithDropout	73	0.999	1.00E-06	FALSE	0.2	0.8	1	0	500	Normal	0.5	0.5	10000	TRUE	[0, 1]	3	0.1657	0.1173	34.8	
TanhWithDropout	73	0.9	1.00E-06	TRUE	0.2	0.2	0	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	2	0.0818	0.0315	58.9	
TanhWithDropout	219	0.9	1.00E-10	TRUE	0.8	0.2	1	0	500	Normal	0.05	2	1	FALSE	[0, 1]	3	0.1944	0.1469	270.5	
TanhWithDropout	73	0.999	1.00E-06	FALSE	0.8	0.2	0	1	500	Uniform	0.05	2	1	FALSE	[-0.5, 0.5]	3	0.1320	0.0910	69.8	
RectifierWithDropout	73	0.9	1.00E-10	FALSE	0.8	0.8	1	1	500	Uniform	0.5	0.5	10000	FALSE	[0, 1]	1	0.1790	0.1474	26.3	
RectifierWithDropout	146	0.9495	1.00E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	89.8	
TanhWithDropout	146	0.9495	1.00E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.4637	0.3660	105.6	
RectifierWithDropout	146	0.9495	1.00E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1741	0.0753	67.4	
TanhWithDropout	146	0.9495	1.00E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1416	0.1386	174.1	
RectifierWithDropout	146	0.9495	1.00E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.7959	0.9878	98.7	

Appendix C: D_ME1DO

Table 20. D_ME1DO Test Matrix

Activation Function	Neurons Per Hidden Layer	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
RectifierWithDropout	219	0.0005	0	0	0.75	0.9	500	FALSE	0.2	0.8	0	1	5	Uniform Adaptive	0.5	0.5	10000	YES	[0, 1]	2	0.1612	0.1168	122.1	
TanhWithDropout	219	0.05	0.0001	2	0.75	0.999	50000	FALSE	0.2	0.8	0	0	500	Normal	0.05	2	1	NO	[0, 1]	2	0.0963	0.0449	101.4	
TanhWithDropout	219	0.05	0	2	0.75	0.999	50000	FALSE	0.8	0.8	1	0	5	Uniform	0.5	2	1	YES	[-1, 1]	1	0.5668	0.4733	64.0	
TanhWithDropout	73	0.0005	0	2	0.75	0.999	500	TRUE	0.8	0.2	0	1	5	Normal	0.5	0.5	1	NO	[-1, 1]	2	0.4426	0.2455	37.6	
TanhWithDropout	73	0.0005	0.0001	2	0.25	0.999	500	TRUE	0.2	0.8	0	0	5	Uniform Adaptive	0.05	0.5	10000	YES	[-1, 1]	1	0.4944	0.2823	32.5	
RectifierWithDropout	219	0.0005	0.0001	0	0.25	0.9	500	FALSE	0.8	0.2	0	0	5	Uniform Adaptive	0.5	2	1	NO	[-0.5, 0.5]	1	0.4254	0.5315	55.2	
TanhWithDropout	73	0.0005	0	0	0.75	0.9	500	TRUE	0.8	0.8	1	0	500	Uniform	0.5	2	10000	NO	[0, 1]	3	0.4038	0.3679	26.2	
RectifierWithDropout	73	0.05	0.0001	2	0.75	0.999	50000	FALSE	0.8	0.8	1	1	5	Uniform Adaptive	0.05	0.5	10000	NO	[0, 1]	3				Exp Grwoth
TanhWithDropout	219	0.0005	0.0001	2	0.75	0.9	50000	FALSE	0.8	0.2	1	1	5	Normal	0.05	2	10000	NO	[0, 1]	1	0.1943	0.1518	75.9	
RectifierWithDropout	73	0.05	0.0001	0	0.75	0.999	50000	TRUE	0.8	0.8	0	0	5	Normal	0.5	0.5	1	NO	[-0.5, 0.5]	3	0.1973	0.2452	13.6	
TanhWithDropout	73	0.05	0.0001	0	0.75	0.9	500	FALSE	0.2	0.2	0	0	5	Normal	0.05	2	10000	YES	[-1, 1]	3	0.9690	1.0076	33.8	
RectifierWithDropout	219	0.0005	0.0001	0	0.75	0.999	500	FALSE	0.2	0.2	1	0	500	Uniform	0.05	0.5	1	NO	[-1, 1]	2	0.7980	0.9878	35.1	
TanhWithDropout	73	0.0005	0.0001	2	0.25	0.9	50000	FALSE	0.2	0.2	0	1	500	Uniform	0.5	0.5	1	YES	[-0.5, 0.5]	3	0.1047	0.0577	194.2	
RectifierWithDropout	219	0.0005	0	0	0.25	0.999	50000	TRUE	0.8	0.2	0	1	500	Normal	0.05	2	10000	YES	[-1, 1]	3	0.7892	0.9810	32.0	
RectifierWithDropout	73	0.05	0.0001	2	0.25	0.999	500	TRUE	0.2	0.2	0	0	500	Uniform	0.5	2	10000	NO	[0, 1]	1				Exp Grwoth
RectifierWithDropout	73	0.0005	0	2	0.75	0.999	500	FALSE	0.8	0.8	0	0	500	Uniform Adaptive	0.05	2	1	YES	[-0.5, 0.5]	3				Exp Grwoth
RectifierWithDropout	219	0.05	0.0001	2	0.75	0.9	50000	TRUE	0.8	0.2	1	0	500	Uniform Adaptive	0.5	0.5	10000	YES	[-1, 1]	2				Exp Grwoth
TanhWithDropout	73	0.05	0.0001	0	0.25	0.999	500	TRUE	0.8	0.2	1	1	5	Uniform Adaptive	0.05	2	1	YES	[0, 1]	2	0.1880	0.1315	35.0	
TanhWithDropout	219	0.0005	0	2	0.25	0.9	50000	TRUE	0.8	0.8	0	0	500	Uniform Adaptive	0.05	0.5	1	NO	[0, 1]	3	0.1058	0.0510	78.6	
RectifierWithDropout	219	0.05	0	2	0.25	0.9	500	TRUE	0.2	0.2	1	0	5	Normal	0.5	0.5	1	YES	[0, 1]	3				Exp Grwoth
RectifierWithDropout	73	0.05	0	2	0.25	0.9	500	FALSE	0.8	0.8	1	1	500	Normal	0.05	0.5	1	NO	[-1, 1]	1				Exp Grwoth
RectifierWithDropout	73	0.0005	0	2	0.25	0.9	50000	TRUE	0.2	0.8	1	0	5	Normal	0.05	2	10000	NO	[-0.5, 0.5]	2				Exp Grwoth
TanhWithDropout	73	0.05	0	0	0.75	0.9	50000	TRUE	0.2	0.2	1	0	500	Uniform Adaptive	0.05	0.5	1	YES	[-0.5, 0.5]	1	0.1951	0.2264	93.2	
TanhWithDropout	219	0.05	0	2	0.25	0.999	500	FALSE	0.2	0.2	1	1	500	Uniform Adaptive	0.5	2	10000	NO	[-0.5, 0.5]	3	0.2764	0.3532	94.2	
RectifierWithDropout	73	0.05	0	0	0.25	0.999	50000	FALSE	0.8	0.2	0	0	5	Uniform	0.05	0.5	10000	YES	[0, 1]	2	0.2002	0.1524	9.2	
RectifierWithDropout	219	0.0005	0.0001	0	0.25	0.9	50000	TRUE	0.2	0.8	1	1	5	Uniform	0.05	2	1	NO	[-1, 1]	3	0.7971	0.9907	31.1	
RectifierWithDropout	219	0.05	0.0001	2	0.75	0.9	500	TRUE	0.8	0.8	0	1	500	Uniform	0.05	2	1	YES	[-0.5, 0.5]	2				Exp Grwoth
RectifierWithDropout	73	0.0005	0.0001	0	0.75	0.999	50000	TRUE	0.2	0.8	1	1	500	Normal	0.5	2	1	YES	[0, 1]	1	0.2021	0.1567	35.4	
TanhWithDropout	219	0.05	0	0	0.75	0.999	500	TRUE	0.2	0.8	0	1	5	Uniform	0.05	0.5	10000	NO	[-0.5, 0.5]	1	0.1941	0.1751	126.9	
TanhWithDropout	219	0.0005	0.0001	0	0.25	0.999	500	FALSE	0.8	0.8	1	0	500	Normal	0.5	0.5	10000	YES	[-0.5, 0.5]	2	0.5994	0.5091	56.9	
TanhWithDropout	73	0.05	0	0	0.25	0.9	50000	FALSE	0.2	0.8	0	1	500	Uniform Adaptive	0.5	2	1	NO	[-1, 1]	2	0.7451	0.9155	51.0	
RectifierWithDropout	73	0.02525	0.00005	1	0.5	0.9495	500	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	1	NO	[-0.5, 0.5]	1				Exp Grwoth
TanhWithDropout	219	0.02525	0.00005	1	0.5	0.9495	50000	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	10000	YES	[-1, 1]	2	0.4452	0.3373	86.0	
RectifierWithDropout	73	0.02525	0.00005	1	0.5	0.9495	500	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	1	NO	[0, 1]	3				Exp Grwoth
TanhWithDropout	219	0.02525	0.00005	1	0.5	0.9495	50000	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	10000	YES	[-0.5, 0.5]	1	0.1425	0.1408	88.7	
RectifierWithDropout	73	0.02525	0.00005	1	0.5	0.9495	500	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	1	NO	[-1, 1]	2				Exp Grwoth

Appendix D: D_ME1AR

Table 21. D_ME1AR Test Matrix

Activation Function	Neurons Per Hidden Layer	Rho	Epsilon	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Tanh	219	0.9	1.00E-10	TRUE	1	1	500	Normal	0.5	0.5	10000	TRUE	[0, 1]	2	0.1952	0.1474	173.5	
Rectifier	73	0.9	1.00E-10	TRUE	1	1	5	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	1	0.1411	0.0918	26.6	
Rectifier	219	0.999	1.00E-06	TRUE	0	0	5	Uniform	0.05	0.5	10000	TRUE	[0, 1]	2				Exp Growth
Rectifier	219	0.999	1.00E-06	TRUE	0	1	5	Normal	0.5	2	10000	FALSE	[-0.5, 0.5]	3	0.1973	0.2452	333.3	
Tanh	219	0.9	1.00E-06	TRUE	1	0	5	Uniform Adaptive	0.5	2	1	FALSE	[-0.5, 0.5]	2	0.1835	0.2240	252.4	
Rectifier	73	0.9	1.00E-10	FALSE	0	0	5	Normal	0.05	2	10000	TRUE	[-0.5, 0.5]	2	0.2084	0.2445	96.0	
Tanh	219	0.999	1.00E-10	TRUE	0	0	500	Normal	0.05	0.5	1	TRUE	[-0.5, 0.5]	1	0.9833	0.1138	175.3	
Tanh	73	0.9	1.00E-10	FALSE	1	1	5	Uniform	0.5	0.5	1	TRUE	[-0.5, 0.5]	3	0.1973	0.2452	78.2	
Rectifier	73	0.9	1.00E-06	FALSE	0	1	500	Normal	0.5	0.5	1	FALSE	[0, 1]	2	0.5075	0.0456	52.4	
Tanh	219	0.999	1.00E-10	FALSE	1	0	5	Normal	0.5	2	1	FALSE	[0, 1]	1	0.1919	0.1493	189.3	
Rectifier	73	0.999	1.00E-06	FALSE	1	0	500	Normal	0.5	2	1	TRUE	[-1, 1]	3	5.0486	0.9810	114.9	
Tanh	219	0.999	1.00E-10	FALSE	0	1	5	Uniform	0.05	2	1	FALSE	[-1, 1]	2	0.3047	0.1248	408.3	
Rectifier	219	0.9	1.00E-06	TRUE	0	1	500	Uniform	0.5	2	1	TRUE	[-0.5, 0.5]	1	343.0929	425.1880	251.3	
Tanh	73	0.9	1.00E-06	TRUE	1	1	5	Normal	0.05	0.5	10000	FALSE	[-1, 1]	1	0.5712	0.5733	120.8	
Rectifier	219	0.9	1.00E-10	TRUE	1	0	5	Uniform	0.5	2	10000	TRUE	[-1, 1]	3	0.7872	0.9713	416.9	
Tanh	73	0.999	1.00E-10	TRUE	0	1	500	Uniform Adaptive	0.5	2	10000	TRUE	[-1, 1]	2	0.3385	0.1622	144.3	
Tanh	219	0.9	1.00E-06	FALSE	0	0	500	Uniform Adaptive	0.05	2	10000	TRUE	[0, 1]	3	0.1366	0.0551	132.7	
Rectifier	73	0.999	1.00E-10	TRUE	1	0	500	Uniform	0.05	2	10000	FALSE	[-0.5, 0.5]	2	0.1987	0.2490	188.4	
Rectifier	73	0.999	1.00E-10	TRUE	0	0	5	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	3	0.0951	0.0087	88.9	
Tanh	73	0.9	1.00E-06	TRUE	0	0	500	Uniform	0.05	0.5	1	FALSE	[-1, 1]	3	0.8398	0.0267	79.6	
Rectifier	219	0.999	1.00E-10	FALSE	1	1	500	Uniform Adaptive	0.05	0.5	10000	FALSE	[-0.5, 0.5]	3	0.5335	0.6877	227.7	
Tanh	73	0.999	1.00E-06	FALSE	1	1	500	Uniform	0.5	2	10000	FALSE	[0, 1]	1	0.1763	0.1306	55.4	
Rectifier	219	0.9	1.00E-10	FALSE	0	0	500	Uniform Adaptive	0.5	0.5	10000	FALSE	[-1, 1]	1	1.7077	2.1248	192.3	
Tanh	73	0.999	1.00E-06	FALSE	0	0	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[-0.5, 0.5]	1	0.2684	0.3105	47.8	
Rectifier	219	0.999	1.00E-06	FALSE	1	1	5	Uniform Adaptive	0.05	0.5	1	TRUE	[-1, 1]	2				Exp Growth
Tanh	146	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1478	0.1517	195.8	
Rectifier	146	0.9495	1.00E-08	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	1038.7930	1244.1760	274.2	
Tanh	146	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1688	0.1186	106.0	
Rectifier	146	0.9495	1.00E-08	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.4310	0.5247	221.6	
Tanh	146	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.4577	0.3432	157.4	

Appendix E: D_ME1

Table 22. D_ME1 Test Matrix

Activation Function	Neurons Per Hidden Layer	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Rectifier	219	0.05	0	0	0.75	0.9	500	FALSE	0	0	500	Uniform	0.05	2	10000	FALSE	[0, 1]	2	0.2002	0.1524	21.4	
Tanh	73	0.0005	0	2	0.75	0.9	50000	TRUE	1	0	500	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	1	0.1907	0.1481	42.4	
Tanh	73	0.05	0	2	0.25	0.999	500	FALSE	0	0	5	Uniform	0.05	0.5	10000	FALSE	[-0.5, 0.5]	3	0.7330	0.5845	33.7	
Tanh	73	0.05	0.0001	2	0.75	0.999	50000	TRUE	1	1	5	Uniform	0.05	2	1	TRUE	[0, 1]	2	0.1799	0.1317	39.7	
Tanh	219	0.05	0.0001	0	0.25	0.9	50000	FALSE	0	1	5	Uniform Adaptive	0.5	2	10000	FALSE	[0, 1]	3	0.1213	0.0662	90.4	
Tanh	219	0.05	0	0	0.25	0.9	50000	FALSE	1	0	5	Uniform Adaptive	0.05	0.5	1	TRUE	[-1, 1]	2	0.6703	0.7057	97.6	
Rectifier	73	0.0005	0.0001	2	0.75	0.9	50000	FALSE	0	0	5	Normal	0.5	0.5	1	FALSE	[-0.5, 0.5]	2				Exp Growth
Tanh	73	0.0005	0	0	0.75	0.9	500	FALSE	1	1	5	Uniform	0.5	2	1	FALSE	[-1, 1]	1	0.9951	0.9208	92.7	
Tanh	219	0.0005	0.0001	2	0.25	0.9	500	TRUE	0	0	500	Uniform	0.5	2	10000	TRUE	[-1, 1]	2	1.2134	0.1351	112.5	
Rectifier	73	0.0005	0	2	0.75	0.999	50000	FALSE	0	1	500	Uniform Adaptive	0.5	2	10000	TRUE	[-1, 1]	3				Exp Growth
Tanh	73	0.05	0.0001	0	0.75	0.9	500	TRUE	0	1	500	Normal	0.05	0.5	1	TRUE	[-1, 1]	3	1.5958	1.6597	34.0	
Rectifier	219	0.0005	0	2	0.25	0.9	500	TRUE	1	1	5	Uniform	0.5	0.5	1	TRUE	[0, 1]	3				Exp Growth
Rectifier	219	0.0005	0	0	0.75	0.999	500	TRUE	0	0	5	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	1	0.1527	0.0827	41.8	
Rectifier	73	0.05	0.0001	0	0.75	0.9	500	TRUE	1	0	5	Uniform Adaptive	0.5	2	10000	TRUE	[-0.5, 0.5]	3	0.1973	0.2452	14.1	
Tanh	73	0.0005	0.0001	2	0.25	0.999	500	FALSE	1	0	500	Normal	0.05	2	1	FALSE	[0, 1]	3	0.1946	0.1471	38.6	
Tanh	73	0.0005	0	0	0.25	0.9	50000	TRUE	0	1	5	Normal	0.05	2	10000	TRUE	[-0.5, 0.5]	1	0.3712	0.2800	53.2	
Tanh	219	0.05	0.0001	2	0.75	0.999	500	FALSE	1	0	5	Normal	0.5	0.5	10000	TRUE	[-1, 1]	1	0.5472	0.5238	129.4	
Tanh	219	0.0005	0.0001	0	0.75	0.999	50000	FALSE	1	1	500	Uniform	0.05	0.5	10000	TRUE	[-0.5, 0.5]	3	0.9482	0.1752	98.0	
Tanh	219	0.0005	0	0	0.25	0.999	50000	TRUE	1	0	500	Normal	0.5	2	1	FALSE	[-0.5, 0.5]	3	0.8846	0.7999	92.6	
Rectifier	73	0.05	0	0	0.25	0.999	500	FALSE	1	1	500	Normal	0.5	0.5	10000	TRUE	[0, 1]	2				Exp Growth
Rectifier	219	0.05	0	2	0.75	0.9	50000	TRUE	1	1	5	Normal	0.05	2	10000	FALSE	[-1, 1]	3				Exp Growth
Rectifier	73	0.0005	0.0001	0	0.25	0.999	500	TRUE	1	1	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[-1, 1]	2	0.8532	1.1193	19.1	
Rectifier	219	0.05	0.0001	2	0.25	0.9	500	FALSE	1	1	500	Uniform Adaptive	0.05	2	1	TRUE	[-0.5, 0.5]	1				Exp Growth
Rectifier	73	0.05	0.0001	0	0.25	0.999	50000	TRUE	0	0	500	Uniform	0.5	0.5	1	FALSE	[-1, 1]	1				Exp Growth
Tanh	219	0.05	0	2	0.75	0.999	500	TRUE	0	1	500	Uniform Adaptive	0.5	0.5	1	FALSE	[-0.5, 0.5]	2	0.2808	0.3514	92.1	
Tanh	146	0.02525	0.00005	1	0.5	0.9495	5000	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1372	0.1313	74.3	
Rectifier	146	0.02525	0.00005	1	0.5	0.9495	5000	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2				Exp Growth
Tanh	146	0.02525	0.00005	1	0.5	0.9495	5000	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1580	0.1069	53.9	
Rectifier	146	0.02525	0.00005	1	0.5	0.9495	5000	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1				Exp Growth
Tanh	146	0.02525	0.00005	1	0.5	0.9495	5000	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.5155	0.4084	156.6	
Rectifier	219	0.0005	0.0001	2	0.25	0.999	50000	FALSE	0	1	5	Normal	0.05	0.5	10000	FALSE	[0, 1]	1				Exp Growth
Rectifier	73	0.05	0	2	0.25	0.999	50000	FALSE	0	0	5	Uniform Adaptive	0.05	2	1	TRUE	[-1, 1]	3				Exp Growth

Appendix F: D_ME2DOAR

Table 23. D_ME2DOAR Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Rho	Epsilon	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate 1	Hidden Dropout Rate 2	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
TanhWithDropout	73	73	0.9	1.00E-10	FALSE	0.2	0.8	0.2	0	1	5	Uniform Adaptive	0.5	0.5	10000	FALSE	[0, 1]	2	0.1218	0.0707	114.4	
TanhWithDropout	73	73	0.999	1E-10	FALSE	0.2	0.2	0.2	0	0	500	Uniform Adaptive	0.5	2	1	TRUE	[-0.5, 0.5]	3	0.1132	0.0613	58.4	
TanhWithDropout	73	219	0.999	1E-10	TRUE	0.8	0.2	0.8	0	1	5	Uniform	0.05	0.5	10000	TRUE	[0, 1]	3	0.1240	0.0693	121.7	
RectifierWithDropout	219	219	0.9	0.000001	FALSE	0.8	0.8	0.2	0	0	5	Normal	0.5	0.5	10000	TRUE	[-0.5, 0.5]	3	0.1973	0.2452	146.2	
TanhWithDropout	73	219	0.9	1E-10	FALSE	0.8	0.8	0.8	1	0	500	Uniform	0.05	2	10000	FALSE	[-0.5, 0.5]	1	0.1607	0.1265	85.6	
TanhWithDropout	219	219	0.999	0.000001	FALSE	0.2	0.8	0.2	1	1	500	Normal	0.5	0.5	10000	FALSE	[0, 1]	1	0.9093	0.8467	158.9	
RectifierWithDropout	73	73	0.999	0.000001	FALSE	0.2	0.8	0.8	1	0	5	Uniform	0.5	0.5	10000	TRUE	[-1, 1]	1	0.7995	0.9707	31.5	
TanhWithDropout	73	73	0.999	0.000001	TRUE	0.2	0.2	0.2	0	0	5	Normal	0.05	2	1	FALSE	[-1, 1]	1	0.5277	0.3071	81.6	
RectifierWithDropout	73	73	0.9	0.000001	FALSE	0.8	0.2	0.2	1	1	500	Uniform	0.5	2	1	TRUE	[0, 1]	2	0.3539	0.4133	18.1	
TanhWithDropout	219	73	0.9	1E-10	TRUE	0.8	0.2	0.8	1	0	500	Normal	0.5	0.5	1	TRUE	[0, 1]	1	0.1516	0.1064	84.7	
TanhWithDropout	73	219	0.9	0.000001	FALSE	0.2	0.2	0.8	1	1	500	Uniform Adaptive	0.5	0.5	1	FALSE	[-1, 1]	3	0.5390	0.5253	135.8	
RectifierWithDropout	73	73	0.9	1E-10	TRUE	0.2	0.2	0.2	1	1	500	Normal	0.05	0.5	10000	FALSE	[-0.5, 0.5]	3	0.1973	0.2452	54.5	
RectifierWithDropout	73	219	0.9	1E-10	TRUE	0.8	0.8	0.2	0	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-1, 1]	1	0.7995	0.9707	119.7	
TanhWithDropout	219	73	0.9	1E-10	TRUE	0.8	0.8	0.2	1	1	5	Uniform	0.5	2	1	FALSE	[-1, 1]	3	0.5577	0.5581	107.0	
RectifierWithDropout	219	73	0.999	1E-10	FALSE	0.2	0.8	0.8	0	1	500	Uniform	0.05	0.5	1	FALSE	[0, 1]	3	0.1482	0.0958	46.5	
RectifierWithDropout	73	219	0.999	1E-10	TRUE	0.2	0.8	0.8	1	1	5	Normal	0.5	2	1	TRUE	[-0.5, 0.5]	2	0.1990	0.2470	82.8	
TanhWithDropout	219	73	0.999	0.000001	FALSE	0.8	0.8	0.8	0	1	500	Normal	0.05	2	10000	TRUE	[-1, 1]	2	0.5532	0.3683	70.3	
RectifierWithDropout	219	219	0.9	1E-10	TRUE	0.2	0.2	0.8	0	0	500	Uniform	0.5	2	10000	FALSE	[-1, 1]	2	0.7960	0.9878	98.7	
RectifierWithDropout	219	219	0.999	1E-10	FALSE	0.8	0.2	0.2	1	0	5	Uniform Adaptive	0.05	0.5	1	FALSE	[-1, 1]	2	0.7960	0.9878	192.1	
TanhWithDropout	219	219	0.9	0.000001	FALSE	0.2	0.2	0.2	0	1	5	Uniform	0.05	2	1	TRUE	[-0.5, 0.5]	1	0.1115	0.0826	354.0	
RectifierWithDropout	219	219	0.999	0.000001	TRUE	0.2	0.8	0.2	1	0	500	Uniform Adaptive	0.05	2	10000	TRUE	[0, 1]	3	0.2015	0.1541	36.7	
TanhWithDropout	219	73	0.9	0.000001	TRUE	0.2	0.8	0.8	1	0	5	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 0.5]	2	0.1977	0.2449	129.0	
RectifierWithDropout	73	219	0.9	0.000001	FALSE	0.8	0.8	0.8	0	0	5	Normal	0.05	2	1	FALSE	[0, 1]	3	0.1285	0.0743	24.7	
TanhWithDropout	73	219	0.999	0.000001	TRUE	0.8	0.8	0.2	0	0	500	Uniform	0.5	0.5	1	FALSE	[-0.5, 0.5]	2	0.7441	0.6657	200.5	
RectifierWithDropout	219	73	0.999	0.000001	TRUE	0.8	0.2	0.8	0	1	5	Uniform Adaptive	0.5	2	10000	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	56.4	
RectifierWithDropout	73	146	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	74.4	
TanhWithDropout	146	219	0.9495	1E-08	TRUE	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.5031	0.4075	147.1	
RectifierWithDropout	146	73	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1378	0.0843	25.8	
TanhWithDropout	219	146	0.9495	1E-08	TRUE	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1411	0.1366	164.2	
RectifierWithDropout	146	146	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.7960	0.9878	89.0	

Appendix G: D_ME2DO

Table 24. D_ME2DO Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate 1	Hidden Dropout Rate 2	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
RectifierWithDropout	219	219	0.05	0	0	0.25	0.9	500	YES	0.8	0.8	0.8	0	0	5	Uniform	0.5	0.5	1	TRUE	[0, 1]	2	0.2080	0.2004	89.0	
RectifierWithDropout	73	219	0.0005	0.0001	0	0.75	0.999	500	NO	0.2	0.8	0.2	1	1	5	Uniform	0.05	2	1	TRUE	[-1, 1]	2	3.3649	3.8718	85.1	
TanhWithDropout	73	219	0.05	0.0001	0	0.75	0.9	50000	YES	0.2	0.8	0.8	1	1	500	Uniform	0.05	0.5	1	TRUE	[0, 1]	3	0.4961	0.4367	73.4	
RectifierWithDropout	73	219	0.0005	0.0001	2	0.25	0.9	50000	NO	0.8	0.8	0.8	1	1	5	Uniform Adaptive	0.5	0.5	1	FALSE	[-0.5, 0.5]	3				Exp Growth
TanhWithDropout	219	73	0.0005	0.0001	0	0.75	0.999	50000	NO	0.8	0.8	0.8	1	0	500	Normal	0.05	2	10000	FALSE	[0, 1]	2	0.6078	0.5537	42.2	
TanhWithDropout	219	73	0.0005	0	2	0.75	0.999	500	NO	0.8	0.8	0.8	0	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[-1, 1]	3	0.5544	0.3587	59.1	
RectifierWithDropout	219	219	0.05	0	2	0.75	0.999	50000	NO	0.8	0.2	0.8	1	1	5	Uniform Adaptive	0.5	2	1	TRUE	[0, 1]	1				Exp Growth
TanhWithDropout	219	73	0.0005	0.0001	2	0.25	0.9	500	NO	0.8	0.2	0.2	1	0	5	Uniform	0.05	0.5	1	FALSE	[-1, 1]	1	0.5670	0.5435	64.1	
TanhWithDropout	73	73	0.0005	0	2	0.25	0.999	50000	YES	0.2	0.8	0.2	0	0	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	3	0.1464	0.0918	51.4	
TanhWithDropout	219	73	0.05	0	0	0.75	0.9	50000	NO	0.8	0.2	0.2	1	1	5	Uniform	0.05	0.5	10000	FALSE	[0, 1]	3	0.5201	0.4673	172.7	
RectifierWithDropout	219	73	0.05	0.0001	2	0.75	0.9	50000	YES	0.2	0.2	0.8	0	1	500	Uniform Adaptive	0.05	0.5	1	FALSE	[-1, 1]	2				Exp Growth
RectifierWithDropout	219	219	0.0005	0.0001	0	0.25	0.9	50000	NO	0.2	0.2	0.8	0	0	5	Normal	0.5	2	10000	TRUE	[-1, 1]	3	1.1031	1.2566	49.7	
TanhWithDropout	73	219	0.05	0.0001	2	0.25	0.999	500	NO	0.8	0.2	0.8	0	1	500	Uniform	0.5	2	10000	FALSE	[-0.5, 0.5]	2	0.1166	0.0734	121.4	
RectifierWithDropout	73	73	0.0005	0.0001	0	0.25	0.999	500	YES	0.8	0.2	0.2	1	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	2	0.2019	0.1733	22.6	
RectifierWithDropout	73	73	0.0005	0	2	0.25	0.9	500	NO	0.2	0.8	0.8	0	1	5	Normal	0.05	2	1	FALSE	[0, 1]	2				Exp Growth
TanhWithDropout	219	219	0.0005	0	0	0.25	0.999	50000	YES	0.2	0.2	0.2	0	1	5	Uniform Adaptive	0.05	2	10000	FALSE	[-0.5, 0.5]	2	0.1587	0.1573	201.4	
RectifierWithDropout	219	73	0.05	0.0001	0	0.75	0.999	500	YES	0.8	0.8	0.2	0	1	5	Normal	0.5	2	1	FALSE	[-0.5, 0.5]	3	2.0879	2.2916	38.8	
TanhWithDropout	73	73	0.05	0.0001	2	0.25	0.999	50000	YES	0.2	0.8	0.8	1	0	5	Uniform	0.5	2	10000	TRUE	[-1, 1]	1				Exp Growth
RectifierWithDropout	73	219	0.05	0	0	0.25	0.999	500	YES	0.8	0.2	0.8	1	0	500	Normal	0.05	2	1	FALSE	[-1, 1]	3				Exp Growth
RectifierWithDropout	73	73	0.0005	0	0	0.75	0.999	50000	NO	0.2	0.2	0.8	0	0	500	Uniform	0.5	0.5	1	FALSE	[-0.5, 0.5]	1	0.8997	1.0413	19.0	
RectifierWithDropout	73	219	0.05	0.0001	2	0.75	0.999	500	NO	0.2	0.2	0.2	0	0	5	Normal	0.05	0.5	10000	TRUE	[0, 1]	3				Exp Growth
RectifierWithDropout	73	73	0.05	0	2	0.75	0.9	50000	YES	0.8	0.2	0.2	1	0	5	Normal	0.5	0.5	10000	TRUE	[-0.5, 0.5]	2				Exp Growth
RectifierWithDropout	219	73	0.05	0.0001	0	0.25	0.999	500	NO	0.2	0.8	0.8	1	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[-0.5, 0.5]	1	0.2350	0.2916	48.4	
TanhWithDropout	219	219	0.05	0	2	0.75	0.9	500	NO	0.2	0.8	0.2	1	0	500	Uniform Adaptive	0.5	2	1	TRUE	[-0.5, 0.5]	2				Exp Growth
RectifierWithDropout	219	73	0.0005	0	2	0.25	0.9	500	YES	0.2	0.2	0.8	1	1	500	Uniform	0.05	2	10000	TRUE	[-0.5, 0.5]	3				Exp Growth
TanhWithDropout	219	219	0.0005	0	2	0.25	0.999	50000	YES	0.8	0.8	0.2	0	1	500	Normal	0.05	0.5	1	TRUE	[-0.5, 0.5]	1	0.1306	0.0727	86.4	
TanhWithDropout	73	73	0.0005	0.0001	0	0.75	0.9	500	YES	0.8	0.2	0.8	0	0	5	Uniform Adaptive	0.05	2	1	TRUE	[-0.5, 0.5]	1	0.1934	0.2315	33.8	
RectifierWithDropout	219	219	0.0005	0.0001	2	0.75	0.9	500	YES	0.8	0.8	0.2	0	0	500	Uniform	0.5	2	10000	FALSE	[0, 1]	1				Exp Growth
RectifierWithDropout	73	73	0.05	0	0	0.25	0.9	50000	NO	0.8	0.8	0.2	0	1	500	Uniform Adaptive	0.05	2	10000	TRUE	[-1, 1]	1	1.1055	1.3403	29.1	
TanhWithDropout	73	219	0.0005	0	0	0.75	0.9	500	YES	0.2	0.8	0.8	1	1	5	Normal	0.5	0.5	10000	FALSE	[-1, 1]	1	0.7534	0.8925	65.6	
TanhWithDropout	219	73	0.05	0.0001	0	0.25	0.9	500	NO	0.2	0.2	0.2	1	1	500	Normal	0.5	2	1	TRUE	[0, 1]	1	0.9379	0.8907	136.7	
RectifierWithDropout	73	146	0.02525	0.00005	1	0.5	0.9495	10000	NO	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1				Exp Growth
TanhWithDropout	146	73	0.02525	0.00005	1	0.5	0.9495	10000	YES	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.5102	0.4059	81.1	
RectifierWithDropout	146	73	0.02525	0.00005	1	0.5	0.9495	10000	NO	0.5	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3				Exp Growth
TanhWithDropout	73	146	0.02525	0.00005	1	0.5	0.9495	10000	YES	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1412	0.1400	55.0	
RectifierWithDropout	146	146	0.02525	0.00005	1	0.5	0.9495	10000	NO	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2				Exp Growth

Appendix H: D_ME2AR

Table 25. D_ME2AR Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Rho	Epsilon	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Rectifier	73	219	0.9	1.00E-10	FALSE	0	1	500	Normal	0.5	0.5	10000	FALSE	[-1, 1]	3	0.7892	1.3015	519.4	
Tanh	219	73	0.999	0.000001	TRUE	0	1	5	Uniform	0.05	2	10000	TRUE	[-0.5, 0.5]	3	0.1116	0.0782	193.2	
Tanh	73	219	0.9	1.00E-10	TRUE	1	1	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[-0.5, 0.5]	2	0.1990	0.2469	267.9	
Tanh	219	73	0.9	0.000001	FALSE	1	1	5	Uniform	0.5	2	10000	FALSE	[-1, 1]	3	0.5621	0.5920	212.9	
Tanh	73	73	0.999	1.00E-10	FALSE	0	1	500	Uniform	0.5	0.5	1	TRUE	[0, 1]	1	0.1303	0.0570	80.8	
Tanh	73	219	0.999	0.000001	TRUE	1	1	500	Normal	0.05	0.5	1	FALSE	[-1, 1]	3	0.9009	0.6980	78.5	
Rectifier	219	73	0.9	1.00E-10	TRUE	0	0	5	Normal	0.5	0.5	1	FALSE	[-0.5, 0.5]	3	0.1973	0.2452	185.8	
Rectifier	73	219	0.9	0.000001	FALSE	0	0	5	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	3	0.1391	0.1243	38.6	
Tanh	73	73	0.999	1.00E-10	TRUE	1	0	500	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	3	0.1940	0.1465	94.7	
Tanh	219	219	0.9	0.000001	FALSE	1	1	500	Normal	0.5	2	1	TRUE	[-0.5, 0.5]	2	0.1971	0.2450	186.5	
Rectifier	219	219	0.999	1.00E-10	TRUE	0	1	5	Normal	0.5	2	10000	TRUE	[0, 1]	1	0.1065	0.0487	233.3	
Tanh	219	219	0.999	1.00E-10	TRUE	0	0	500	Uniform Adaptive	0.05	2	10000	FALSE	[-1, 1]	2	0.7322	0.0025	539.8	
Rectifier	219	219	0.999	1.00E-10	FALSE	1	0	500	Uniform	0.05	0.5	10000	TRUE	[-0.5, 0.5]	3	61.4845	0.2452	936.1	
Rectifier	73	73	0.9	0.000001	TRUE	0	0	500	Uniform	0.5	2	10000	TRUE	[-1, 1]	2	0.7960	0.9878	101.8	
Tanh	219	73	0.9	0.000001	TRUE	1	0	500	Normal	0.05	0.5	10000	TRUE	[0, 1]	1	0.1883	0.1458	106.1	
Rectifier	73	73	0.9	0.000001	TRUE	1	1	500	Uniform Adaptive	0.05	0.5	10000	TRUE	[-1, 1]	1	0.7995	0.9707	121.2	
Rectifier	219	73	0.999	0.000001	FALSE	1	1	5	Uniform Adaptive	0.5	0.5	10000	FALSE	[0, 1]	2	0.1784	0.1302	20.6	
Rectifier	73	219	0.999	0.000001	TRUE	1	0	5	Uniform	0.5	2	1	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	162.3	
Tanh	73	73	0.999	0.000001	FALSE	0	0	5	Normal	0.05	0.5	10000	FALSE	[-0.5, 0.5]	2	0.3265	0.2865	55.2	
Rectifier	73	73	0.999	1.00E-10	FALSE	1	1	5	Normal	0.05	2	1	TRUE	[-1, 1]	2	1.1760	1.0003	122.9	
Tanh	219	219	0.999	0.000001	FALSE	0	0	5	Uniform Adaptive	0.5	0.5	1	TRUE	[-1, 1]	1	0.8793	0.6679	90.1	
Rectifier	219	73	0.9	1.00E-10	FALSE	0	1	500	Uniform Adaptive	0.05	2	1	FALSE	[-0.5, 0.5]	1	0.2003	0.2427	348.6	
Rectifier	219	219	0.9	0.000001	TRUE	0	1	500	Uniform	0.05	0.5	1	FALSE	[0, 1]	2	1.7685	0.0514	44.4	
Tanh	73	219	0.9	1.00E-10	FALSE	1	0	5	Uniform	0.05	2	10000	FALSE	[0, 1]	1	0.1921	0.1496	261.0	
Tanh	219	73	0.9	1.00E-10	TRUE	1	0	5	Uniform	0.05	0.5	1	TRUE	[-1, 1]	2	0.5503	0.5414	179.8	
Tanh	146	73	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1479	0.1518	202.7	
Rectifier	219	146	0.9495	1.00E-08	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.7960	0.9878	234.7	
Tanh	73	146	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1676	0.1174	112.0	
Rectifier	146	219	0.9495	1.00E-08	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1999	0.2427	223.7	
Tanh	146	146	0.9495	1.00E-08	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.4645	0.3675	324.3	

Appendix I: D_ME2

Table 26. D_ME2 Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Tanh	73	73	0.05	0.0001	2	0.75	0.9	500	TRUE	0	1	500	Normal	0.5	2	10000	FALSE	[-1, 1]	1	0.5849	0.3710	99.0	
Rectifier	219	73	0.0005	0	0	0.75	0.9	50000	TRUE	1	1	5	Normal	0.05	0.5	10000	TRUE	[0, 1]	1	2.3330	2.4207	35.9	
Tanh	219	73	0.0005	0	2	0.75	0.9	50000	FALSE	1	1	5	Uniform	0.5	2	10000	TRUE	[-0.5, 0.5]	3	0.1970	0.2449	94.3	
Tanh	73	73	0.0005	0	2	0.75	0.999	50000	FALSE	0	0	500	Uniform Adaptive	0.05	2	1	TRUE	[-1, 1]	2	0.8703	1.0864	117.3	
Rectifier	219	219	0.0005	0	0	0.75	0.9	500	TRUE	0	0	500	Uniform Adaptive	0.5	2	1	TRUE	[-1, 1]	1	1.5367	1.9243	177.0	
Rectifier	219	219	0.0005	0.0001	2	0.25	0.9	50000	FALSE	0	0	5	Uniform	0.05	0.5	10000	FALSE	[-1, 1]	1				Exp Growth
Rectifier	73	219	0.0005	0	2	0.25	0.999	500	FALSE	1	1	500	Normal	0.5	0.5	1	TRUE	[-1, 1]	3				Exp Growth
Tanh	219	219	0.0005	0.0001	2	0.25	0.999	50000	TRUE	0	1	500	Normal	0.05	0.5	10000	TRUE	[0, 1]	2	0.1806	0.0527	476.3	
Tanh	219	73	0.0005	0.0001	2	0.25	0.9	500	TRUE	1	1	500	Uniform	0.05	2	1	FALSE	[-0.5, 0.5]	2	0.1989	0.2469	111.9	
Tanh	73	219	0.0005	0	2	0.75	0.9	500	TRUE	0	1	5	Uniform Adaptive	0.05	0.5	1	FALSE	[0, 1]	1	0.1133	0.0587	93.4	
Tanh	219	73	0.0005	0.0001	0	0.75	0.999	50000	FALSE	0	0	5	Normal	0.5	2	1	FALSE	[0, 1]	3	0.6135	0.1539	65.4	
Rectifier	219	73	0.05	0	0	0.25	0.9	500	TRUE	0	0	5	Normal	0.05	0.5	1	TRUE	[-0.5, 0.5]	3				Exp Growth
Rectifier	219	73	0.05	0	2	0.25	0.9	50000	FALSE	1	0	500	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	2				Exp Growth
Rectifier	219	219	0.05	0.0001	0	0.75	0.999	500	FALSE	1	1	5	Uniform Adaptive	0.05	2	10000	TRUE	[-1, 1]	2	1.6315	1.9952	88.3	
Tanh	73	219	0.05	0	0	0.25	0.999	500	FALSE	1	1	5	Uniform	0.05	2	1	FALSE	[0, 1]	1	0.6568	0.6433	172.4	
Rectifier	73	219	0.0005	0.0001	2	0.75	0.999	500	TRUE	1	0	5	Normal	0.5	2	1	FALSE	[-0.5, 0.5]	2				Exp Growth
Rectifier	219	219	0.05	0	2	0.75	0.999	500	TRUE	0	0	500	Uniform	0.05	2	10000	FALSE	[0, 1]	3				Exp Growth
Rectifier	73	73	0.0005	0.0001	0	0.25	0.9	500	FALSE	1	0	500	Uniform	0.5	2	10000	TRUE	[0, 1]	1	10.6115	10.7605	22.2	
Rectifier	73	73	0.05	0.0001	2	0.75	0.999	50000	TRUE	1	0	500	Uniform	0.05	0.5	1	TRUE	[-0.5, 0.5]	1				Exp Growth
Tanh	73	219	0.0005	0.0001	0	0.25	0.9	50000	TRUE	1	0	5	Uniform Adaptive	0.05	2	1	FALSE	[-1, 1]	3	0.7341	0.8570	116.8	
Rectifier	73	219	0.05	0	0	0.75	0.9	50000	FALSE	0	1	5	Uniform	0.5	0.5	1	FALSE	[-0.5, 0.5]	2	2.5956	2.7942	73.4	
Rectifier	73	73	0.0005	0.0001	0	0.75	0.999	500	FALSE	0	1	500	Uniform Adaptive	0.05	0.5	10000	FALSE	[-0.5, 0.5]	3	0.4655	0.5504	78.2	
Tanh	73	219	0.05	0	2	0.25	0.9	50000	FALSE	0	0	500	Normal	0.05	2	10000	TRUE	[-0.5, 0.5]	1	0.8379	0.7369	141.1	
Rectifier	73	219	0.05	0.0001	0	0.25	0.9	50000	TRUE	0	1	500	Uniform Adaptive	0.5	2	10000	TRUE	[0, 1]	2	0.4443	0.4514	51.4	
Tanh	73	73	0.0005	0	0	0.25	0.999	500	TRUE	0	0	5	Uniform	0.5	0.5	10000	TRUE	[-1, 1]	2	1.4089	1.1992	60.7	
Tanh	219	73	0.05	0	0	0.75	0.9	500	FALSE	1	0	500	Normal	0.05	0.5	10000	FALSE	[-1, 1]	2	1.4902	1.6014	51.6	
Tanh	73	219	0.05	0.0001	2	0.75	0.9	500	FALSE	1	0	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	3	0.1845	0.1366	89.6	
Rectifier	73	73	0.05	0	2	0.25	0.999	50000	TRUE	1	1	5	Uniform Adaptive	0.5	2	10000	FALSE	[-1, 1]	3				Exp Growth
Tanh	219	219	0.05	0.0001	0	0.75	0.9	50000	TRUE	1	1	500	Uniform	0.5	0.5	1	TRUE	[-1, 1]	3	1.5229	1.6427	93.4	
Tanh	219	73	0.05	0.0001	2	0.25	0.999	500	FALSE	0	1	5	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 0.5]	1	0.1468	0.0929	114.4	
Tanh	219	219	0.0005	0	0	0.25	0.999	50000	TRUE	1	0	500	Uniform Adaptive	0.5	0.5	10000	FALSE	[-0.5, 0.5]	1	0.1887	0.2193	309.6	
Tanh	146	73	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1378	0.1320	61.5	
Rectifier	146	73	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2				Exp Growth
Tanh	73	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1592	0.1081	61.1	
Rectifier	73	146	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1				Exp Growth
Tanh	146	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-1, 1]	2	0.5163	0.4091	203.1	

Appendix J: D_ME3DOAR

Table 27. D_ME3DOAR Test Design

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Neurons Per Hidden Layer 3	Rho	Epsilon	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate 1	Hidden Dropout Rate 2	Hidden Dropout Rate 3	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
RectifierWithDropout	73	219	73	0.999	1.00E-10	TRUE	0.2	0.8	0.8	0.8	1	0	500	Uniform	0.5	0.5	1	FALSE	[0, 1]	2	0.1934	0.1456	37.3	
RectifierWithDropout	73	73	73	0.999	0.000001	FALSE	0.8	0.2	0.2	0.8	0	1	500	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 0.5]	2	0.1990	0.2470	41.7	
RectifierWithDropout	219	219	219	0.9	0.000001	TRUE	0.2	0.2	0.8	0.2	1	0	500	Uniform	0.5	2	1	TRUE	[-0.5, 0.5]	1	0.1999	0.2427	156.3	
TanhWithDropout	219	73	73	0.999	1E-10	FALSE	0.8	0.2	0.8	0.2	0	0	5	Uniform	0.05	0.5	1	FALSE	[-1, 1]	3	0.4854	0.2663	186.3	
TanhWithDropout	219	73	73	0.999	0.000001	TRUE	0.2	0.8	0.2	0.2	0	1	500	Uniform	0.05	0.5	10000	TRUE	[0, 1]	1	0.1296	0.0766	123.4	
RectifierWithDropout	73	73	219	0.999	1E-10	FALSE	0.2	0.8	0.2	0.2	0	0	500	Normal	0.05	0.5	10000	TRUE	[-0.5, 0.5]	1	0.1999	0.2427	85.8	
TanhWithDropout	219	73	219	0.9	0.000001	TRUE	0.8	0.2	0.8	0.8	0	0	500	Normal	0.05	2	10000	FALSE	[0, 1]	2	0.1417	0.0914	183.0	
TanhWithDropout	73	219	219	0.9	1E-10	TRUE	0.8	0.8	0.2	0.8	1	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[-0.5, 0.5]	1	0.1998	0.2426	130.9	
TanhWithDropout	219	219	73	0.9	1E-10	FALSE	0.2	0.2	0.8	0.8	1	1	500	Normal	0.05	0.5	1	TRUE	[-0.5, 0.5]	3	0.1432	0.1220	117.9	
TanhWithDropout	73	219	73	0.999	1E-10	TRUE	0.8	0.2	0.8	0.2	0	1	500	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	1	0.1269	0.0727	77.0	
TanhWithDropout	73	73	219	0.9	0.000001	FALSE	0.8	0.8	0.8	0.2	1	1	500	Uniform	0.05	0.5	1	TRUE	[-1, 1]	2	0.5512	0.5569	116.4	
RectifierWithDropout	73	73	219	0.9	1E-10	FALSE	0.2	0.2	0.2	0.2	1	1	500	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	3	0.1966	0.1491	76.7	
RectifierWithDropout	219	219	73	0.9	0.000001	TRUE	0.2	0.2	0.2	0.2	0	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[-0.5, 0.5]	2	0.1990	0.2470	173.4	
TanhWithDropout	219	73	73	0.9	0.000001	FALSE	0.2	0.2	0.2	0.8	1	1	5	Uniform	0.5	0.5	10000	FALSE	[0, 1]	1	0.1858	0.1435	86.7	
RectifierWithDropout	219	219	73	0.999	1E-10	FALSE	0.8	0.8	0.2	0.2	1	1	500	Uniform	0.05	2	10000	FALSE	[-1, 1]	2	0.7960	0.9878	79.6	
TanhWithDropout	219	73	219	0.999	1E-10	TRUE	0.2	0.8	0.2	0.2	0	1	5	Normal	0.5	2	1	FALSE	[-0.5, 0.5]	2	0.1153	0.0690	200.5	
RectifierWithDropout	73	219	73	0.9	0.000001	FALSE	0.2	0.8	0.8	0.8	0	1	5	Normal	0.05	2	1	FALSE	[-1, 1]	1	0.7995	0.9707	31.4	
RectifierWithDropout	219	73	219	0.9	1E-10	FALSE	0.8	0.2	0.2	0.8	0	0	5	Uniform	0.05	2	1	TRUE	[0, 1]	1	0.1436	0.0892	35.2	
RectifierWithDropout	73	73	73	0.999	1E-10	FALSE	0.2	0.2	0.8	0.8	1	0	5	Normal	0.05	2	10000	TRUE	[0, 1]	2	0.1825	0.1361	15.2	
RectifierWithDropout	73	73	73	0.9	1E-10	TRUE	0.8	0.8	0.8	0.8	0	1	5	Uniform	0.5	2	10000	TRUE	[-0.5, 0.5]	3	0.1972	0.2450	41.8	
TanhWithDropout	73	219	73	0.9	0.000001	FALSE	0.8	0.8	0.2	0.2	0	0	500	Normal	0.5	0.5	1	FALSE	[0, 1]	3	0.1517	0.0983	84.7	
TanhWithDropout	73	219	219	0.999	0.000001	FALSE	0.2	0.2	0.2	0.8	0	1	500	Uniform	0.05	2	10000	FALSE	[-0.5, 0.5]	3	0.1150	0.0848	121.0	
TanhWithDropout	73	219	219	0.999	1E-10	TRUE	0.2	0.2	0.2	0.8	0	0	5	Uniform	0.5	0.5	1	TRUE	[-1, 1]	2	0.4669	0.2605	136.6	
RectifierWithDropout	219	219	219	0.999	0.000001	TRUE	0.8	0.8	0.2	0.8	1	1	5	Normal	0.05	0.5	1	TRUE	[0, 1]	3	0.1691	0.1214	38.1	
RectifierWithDropout	219	73	219	0.9	1E-10	TRUE	0.2	0.8	0.8	0.8	0	1	500	Uniform Adaptive	0.05	0.5	1	FALSE	[-1, 1]	1	0.7995	0.9707	62.4	
RectifierWithDropout	73	73	219	0.999	0.000001	TRUE	0.8	0.2	0.8	0.2	1	1	5	Normal	0.5	0.5	10000	FALSE	[-1, 1]	1	0.7995	0.9707	110.5	
TanhWithDropout	219	219	73	0.9	1E-10	TRUE	0.8	0.2	0.2	0.8	1	0	500	Normal	0.5	2	10000	TRUE	[-1, 1]	1	0.5172	0.4839	80.1	
RectifierWithDropout	219	219	219	0.999	0.000001	FALSE	0.2	0.8	0.8	0.8	0	0	500	Uniform Adaptive	0.5	2	10000	TRUE	[-1, 1]	3	0.7892	0.9810	79.3	
TanhWithDropout	219	73	73	0.999	0.000001	FALSE	0.8	0.8	0.8	0.8	1	0	5	Uniform Adaptive	0.5	2	1	FALSE	[-0.5, 0.5]	1				Exp Growth
TanhWithDropout	219	219	219	0.9	1E-10	FALSE	0.2	0.8	0.8	0.2	0	1	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	2	0.1213	0.0701	100.7	
TanhWithDropout	73	73	73	0.9	0.000001	TRUE	0.2	0.8	0.2	0.2	1	0	5	Uniform Adaptive	0.05	2	1	TRUE	[-1, 1]	3	0.5543	0.5408	143.9	
RectifierWithDropout	146	73	146	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	75.5	
TanhWithDropout	146	73	146	0.9495	1E-08	TRUE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.5068	0.4063	139.1	
RectifierWithDropout	146	146	146	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	0.1713	0.1159	81.4	
TanhWithDropout	146	146	146	0.9495	1E-08	TRUE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1410	0.1367	106.4	
RectifierWithDropout	73	146	73	0.9495	1E-08	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1	0.1999	0.2427	47.0	
TanhWithDropout	73	146	73	0.9495	1E-08	TRUE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.5072	0.4093	71.9	

Appendix K: D_ME3DO

Table 28. D_ME3DO Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Neurons Per Hidden Layer 3	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate 1	Hidden Dropout Rate 2	Hidden Dropout Rate 3	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
TanhWithDropout	73	219	73	0.05	0.0001	2	0.25	0.999	500	FALSE	0.8	0.2	0.8	0.8	0	1	5	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	1	0.1434	0.0895	152.0	
TanhWithDropout	219	219	219	0.05	0.0001	0	0.25	0.999	50000	FALSE	0.8	0.8	0.2	0.8	1	1	500	Normal	0.05	2	10000	TRUE	[-1, 1]	3	1.2125	1.2576	114.3	
TanhWithDropout	73	73	73	0.0005	0	0	0.75	0.999	50000	FALSE	0.8	0.2	0.2	0.2	0	0	5	Normal	0.5	0.5	1	TRUE	[-1, 1]	3	0.9941	1.0959	42.8	
TanhWithDropout	219	219	73	0.0005	0	2	0.25	0.9	500	FALSE	0.2	0.8	0.2	0.8	0	1	5	Uniform	0.5	0.5	10000	FALSE	[-0.5, 0.5]	3	0.1369	0.0908	101.8	
RectifierWithDropout	73	219	73	0.0005	0.0001	0	0.75	0.9	500	FALSE	0.2	0.2	0.2	0.2	0	1	500	Uniform	0.05	2	1	FALSE	[-1, 1]	1				Exp Growth
RectifierWithDropout	219	219	73	0.05	0	2	0.75	0.9	500	FALSE	0.2	0.2	0.8	0.8	0	0	5	Uniform Adaptive	0.05	0.5	10000	TRUE	[-1, 1]	2				Exp Growth
RectifierWithDropout	73	219	219	0.0005	0.0001	2	0.75	0.9	50000	FALSE	0.2	0.2	0.2	0.2	1	0	5	Uniform Adaptive	0.5	2	10000	FALSE	[-0.5, 0.5]	3				Exp Growth
RectifierWithDropout	219	73	219	0.0005	0.0001	2	0.25	0.999	500	FALSE	0.2	0.2	0.8	0.2	1	1	500	Normal	0.5	0.5	10000	TRUE	[-1, 1]	1				Exp Growth
TanhWithDropout	219	219	219	0.05	0.0001	0	0.75	0.999	500	TRUE	0.2	0.8	0.2	0.8	0	0	5	Normal	0.5	0.5	1	FALSE	[-0.5, 0.5]	2	0.1886	0.2104	114.8	
TanhWithDropout	219	219	73	0.05	0.0001	0	0.25	0.9	500	TRUE	0.2	0.2	0.8	0.2	1	0	5	Normal	0.05	2	1	TRUE	[0, 1]	3	0.2683	0.2389	169.6	
TanhWithDropout	73	73	73	0.0005	0.0001	0	0.75	0.9	50000	TRUE	0.2	0.8	0.8	0.8	0	1	500	Uniform Adaptive	0.05	0.5	10000	TRUE	[-0.5, 0.5]	1	0.1988	0.2415	52.3	
TanhWithDropout	73	219	219	0.0005	0	2	0.25	0.999	50000	TRUE	0.2	0.8	0.8	0.2	0	1	5	Uniform	0.05	2	10000	TRUE	[0, 1]	2	0.1280	0.0770	181.9	
RectifierWithDropout	219	73	73	0.05	0	2	0.75	0.9	50000	TRUE	0.2	0.2	0.8	0.8	0	1	500	Normal	0.5	2	1	FALSE	[0, 1]	3				Exp Growth
RectifierWithDropout	73	73	219	0.05	0	0	0.25	0.999	500	TRUE	0.2	0.8	0.8	0.2	1	1	5	Uniform Adaptive	0.05	0.5	1	FALSE	[-1, 1]	3	1.4734	1.8216	71.0	
RectifierWithDropout	73	73	73	0.05	0.0001	2	0.25	0.9	500	TRUE	0.8	0.2	0.2	0.2	0	1	5	Normal	0.05	0.5	10000	FALSE	[-0.5, 0.5]	2				Exp Growth
RectifierWithDropout	73	219	219	0.05	0	2	0.25	0.9	50000	FALSE	0.8	0.2	0.8	0.8	0	0	500	Uniform	0.05	0.5	1	TRUE	[-0.5, 0.5]	3				Exp Growth
TanhWithDropout	219	219	219	0.05	0	0	0.75	0.9	50000	TRUE	0.8	0.2	0.8	0.2	1	1	5	Uniform	0.5	0.5	10000	FALSE	[-1, 1]	1	0.7843	0.8849	140.8	
TanhWithDropout	73	73	73	0.05	0	0	0.25	0.999	500	FALSE	0.2	0.2	0.8	0.8	1	0	500	Uniform	0.5	2	10000	FALSE	[-0.5, 0.5]	2				Exp Growth
RectifierWithDropout	73	219	73	0.0005	0	0	0.25	0.9	500	TRUE	0.8	0.8	0.8	0.8	1	0	5	Normal	0.5	2	1	TRUE	[-0.5, 0.5]	1	0.3801	0.4558	26.0	
RectifierWithDropout	219	73	219	0.0005	0	0	0.25	0.999	500	TRUE	0.8	0.2	0.2	0.8	0	0	5	Uniform Adaptive	0.05	2	10000	FALSE	[0, 1]	1	0.1722	0.1433	142.4	
RectifierWithDropout	219	73	73	0.05	0	0	0.25	0.9	50000	FALSE	0.8	0.8	0.2	0.2	1	1	5	Uniform Adaptive	0.5	0.5	1	TRUE	[0, 1]	2	0.1943	0.1879	35.4	
TanhWithDropout	73	73	219	0.0005	0	2	0.75	0.9	500	FALSE	0.8	0.8	0.8	0.8	1	1	5	Normal	0.05	2	1	FALSE	[-1, 1]	2	0.5425	0.5417	70.4	
TanhWithDropout	219	219	73	0.0005	0	2	0.75	0.999	500	TRUE	0.8	0.2	0.2	0.2	1	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 0.5]	3	0.1962	0.2439	125.1	
TanhWithDropout	73	73	219	0.05	0	2	0.25	0.9	500	TRUE	0.2	0.8	0.2	0.2	0	0	500	Uniform Adaptive	0.5	2	1	TRUE	[-1, 1]	1				Exp Growth
RectifierWithDropout	73	219	219	0.0005	0.0001	0	0.75	0.9	500	TRUE	0.2	0.2	0.2	0.8	1	1	500	Uniform	0.5	0.5	1	TRUE	[0, 1]	2				Exp Growth
RectifierWithDropout	73	219	73	0.05	0	2	0.75	0.999	50000	FALSE	0.2	0.8	0.2	0.2	1	0	500	Normal	0.05	0.5	10000	FALSE	[0, 1]	1				Exp Growth
RectifierWithDropout	219	73	219	0.05	0.0001	2	0.75	0.999	50000	FALSE	0.2	0.8	0.8	0.2	0	1	5	Uniform	0.05	2	1	TRUE	[-0.5, 0.5]	1				Exp Growth
TanhWithDropout	219	73	219	0.0005	0.0001	0	0.75	0.9	500	FALSE	0.8	0.8	0.8	0.2	0	0	500	Uniform	0.05	0.5	10000	FALSE	[0, 1]	3	0.8759	0.8255	67.2	
RectifierWithDropout	219	219	73	0.0005	0.0001	2	0.25	0.999	50000	TRUE	0.8	0.8	0.8	0.2	0	0	500	Uniform Adaptive	0.5	2	1	FALSE	[-1, 1]	2				Exp Growth
RectifierWithDropout	73	73	73	0.05	0.0001	2	0.75	0.999	500	TRUE	0.8	0.8	0.2	0.8	1	0	5	Uniform	0.5	2	10000	TRUE	[-1, 1]	3				Exp Growth
TanhWithDropout	219	73	73	0.0005	0.0001	2	0.25	0.9	50000	TRUE	0.2	0.2	0.2	0.8	1	0	5	Uniform	0.05	0.5	1	FALSE	[-1, 1]	1	0.5645	0.5406	41.2	
RectifierWithDropout	146	73	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 0.5]	1				Exp Growth
TanhWithDropout	146	73	146	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-1, 1]	2	0.5085	0.4052	81.6	
RectifierWithDropout	73	146	73	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3				Exp Growth
TanhWithDropout	73	146	73	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 0.5]	1	0.1410	0.1370	43.4	
RectifierWithDropout	143	143	143	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3				Exp Growth
TanhWithDropout	143	143	143	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3	0.1633	0.1126	89.7	

Appendix L: D_ME3AR

Table 29. D_ME3AR Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Neurons Per Hidden Layer 3	Rho	Epsilon	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale		Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Tanh	219	73	219	0.9	1.00E-10	FALSE	0	0	5	Normal	0.5	0.5	1	TRUE	ZO	[0, 1]	3	0.1426	0.0241	373.1	
Rectifier	73	73	219	0.999	1E-10	TRUE	0	0	500	Normal	0.5	0.5	10000	TRUE	PM.5	[-0.5, 0.5]	2	0.1990	0.2470	231.5	
Rectifier	73	219	219	0.999	0.000001	TRUE	0	1	5	Normal	0.5	2	10000	FALSE	PM1	[-1, 1]	3	0.7892	0.9810	216.3	
Tanh	219	219	73	0.9	1E-10	TRUE	0	0	5	Normal	0.05	2	10000	TRUE	PM1	[-1, 1]	1	0.7317	0.0312	419.3	
Rectifier	219	219	73	0.999	0.000001	FALSE	1	0	5	Uniform	0.5	0.5	10000	TRUE	PM1	[-1, 1]	3	0.7892	0.9810	135.8	
Rectifier	219	219	73	0.999	1E-10	TRUE	1	1	500	Normal	0.5	0.5	1	FALSE	PM.5	[-0.5, 0.5]	1	3475.3776	0.2431	130.2	
Tanh	219	73	73	0.999	1E-10	TRUE	0	1	5	Uniform Adaptive	0.5	2	10000	FALSE	PM.5	[-0.5, 0.5]	3	0.1171	0.0689	199.8	
Rectifier	73	73	73	0.9	0.000001	TRUE	0	0	500	Uniform Adaptive	0.05	0.5	10000	FALSE	ZO	[0, 1]	3	0.2015	0.1541	33.3	
Rectifier	219	73	73	0.999	0.000001	TRUE	1	0	500	Uniform Adaptive	0.05	2	1	TRUE	PM1	[-1, 1]	2	0.7960	0.9878	157.9	
Tanh	73	219	73	0.9	0.000001	FALSE	1	1	5	Normal	0.05	0.5	10000	FALSE	PM.5	[-0.5, 0.5]	2	0.1974	0.2448	150.8	
Rectifier	219	73	219	0.9	0.000001	TRUE	1	1	5	Uniform	0.05	0.5	1	TRUE	PM.5	[-0.5, 0.5]	3	0.1973	0.2452	296.7	
Rectifier	219	73	219	0.999	0.000001	FALSE	1	0	5	Normal	0.5	2	1	FALSE	ZO	[0, 1]	1				Exp Growth
Tanh	73	219	73	0.999	0.000001	TRUE	0	1	5	Uniform Adaptive	0.5	0.5	1	TRUE	ZO	[0, 1]	2	0.2493	0.2019	113.8	
Rectifier	73	219	73	0.999	1E-10	FALSE	0	1	500	Uniform	0.05	2	1	TRUE	ZO	[0, 1]	3	3.4005	0.1231	75.2	
Tanh	73	73	73	0.9	0.000001	TRUE	1	1	500	Uniform	0.5	2	10000	TRUE	ZO	[0, 1]	1	0.1885	0.1458	145.1	
Tanh	73	73	219	0.999	1E-10	FALSE	1	1	5	Uniform Adaptive	0.05	0.5	10000	TRUE	PM1	[-1, 1]	1	0.5978	0.6197	328.2	
Tanh	219	219	219	0.999	0.000001	FALSE	0	0	500	Uniform	0.05	0.5	10000	FALSE	PM.5	[-0.5, 0.5]	1	0.8275	0.6808	197.6	
Rectifier	73	73	73	0.9	1E-10	FALSE	1	0	5	Uniform	0.5	2	10000	FALSE	PM.5	[-0.5, 0.5]	2	0.1990	0.2470	131.8	
Rectifier	73	219	219	0.9	0.000001	FALSE	0	0	5	Uniform Adaptive	0.5	2	1	TRUE	PM.5	[-0.5, 0.5]	1	0.1999	0.2427	269.8	
Tanh	219	73	219	0.9	0.000001	FALSE	0	1	500	Uniform	0.5	2	1	FALSE	PM1	[-1, 1]	2	0.6421	0.4375	124.6	
Rectifier	73	73	73	0.9	1E-10	FALSE	0	1	5	Uniform	0.05	0.5	1	FALSE	PM1	[-1, 1]	1	0.7995	0.9707	113.0	
Tanh	73	73	73	0.999	0.000001	FALSE	1	0	500	Normal	0.05	2	1	TRUE	PM.5	[-0.5, 0.5]	3	0.1889	0.2325	141.9	
Tanh	73	219	219	0.9	1E-10	TRUE	1	0	500	Uniform Adaptive	0.5	0.5	1	FALSE	PM1	[-1, 1]	3	0.5508	0.5391	252.5	
Rectifier	219	219	219	0.9	1E-10	FALSE	1	1	500	Uniform Adaptive	0.05	2	10000	TRUE	ZO	[0, 1]	2	0.5539	0.5059	51.9	
Tanh	73	219	219	0.999	1E-10	TRUE	1	0	5	Uniform	0.05	2	1	FALSE	ZO	[0, 1]	2	0.1927	0.1449	282.4	
Tanh	146	73	146	0.9495	1E-08	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	PM.5	[-0.5, 0.5]	1	0.1479	0.1518	246.2	
Rectifier	146	146	146	0.9495	1E-08	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	PM1	[-1, 1]	2	0.7960	0.9878	211.0	
Tanh	73	146	73	0.9495	1E-08	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	ZO	[0, 1]	3	0.1683	0.1181	105.2	
Rectifier	146	73	146	0.9495	1E-08	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	PM.5	[-0.5, 0.5]	1	0.1999	0.2427	210.4	
Tanh	146	146	146	0.9495	1E-08	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	PM1	[-1, 1]	2	0.4847	0.3690	292.0	
Rectifier	146	146	146	0.9495	1E-08	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	PM1	[-1, 1]	2	0.7960	0.9878	264.8	

Appendix M: D_ME3

Table 30. D_ME3 Test Matrix

Activation Function	Neurons Per Hidden Layer 1	Neurons Per Hidden Layer 2	Neurons Per Hidden Layer 3	Learning Rate	Rate Annealing	Rate Decay	Momentum Start	Momentum Stable	Momentum Ramp	Nesterov Accelerated Gradient	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale		Test/Training Set	Test MSE	Train MSE	Time (Sec)	Notes
Tanh	73	219	219	0.0005	0	0	0.75	0.999	50000	TRUE	1	0	5	Uniform	0.5	0.5	10000	TRUE	ZO	[0, 1]	3	0.5461	0.4418	219.1	
Tanh	219	219	219	0.0005	0.0001	2	0.75	0.9	500	FALSE	0	0	5	Normal	0.5	2	10000	TRUE	PM.5	[-0.5, 0.5]	1	0.1728	0.0265	262.3	
Tanh	219	73	73	0.05	0.0001	0	0.75	0.9	50000	TRUE	1	0	5	Normal	0.5	2	1	TRUE	PM1	[-1, 1]	2	1.0530	1.1766	133.1	
Rectifier	73	219	73	0.05	0.0001	0	0.25	0.9	50000	TRUE	0	0	500	Uniform	0.05	0.5	1	TRUE	PM.5	[-0.5, 0.5]	3				Exp Growth
Tanh	73	73	219	0.05	0.0001	2	0.25	0.9	50000	FALSE	1	0	500	Uniform Adaptive	0.5	2	10000	TRUE	PM1	[-1, 1]	3				Exp Growth
Rectifier	219	73	73	0.05	0	0	0.25	0.999	500	FALSE	0	0	5	Normal	0.05	2	10000	TRUE	ZO	[0, 1]	3				Exp Growth
Tanh	73	73	73	0.0005	0	0	0.25	0.9	500	TRUE	1	1	500	Uniform Adaptive	0.5	2	10000	TRUE	ZO	[0, 1]	1	0.2041	0.1509	172.9	
Rectifier	73	219	73	0.0005	0	2	0.75	0.999	500	TRUE	1	0	5	Uniform Adaptive	0.05	2	1	TRUE	PM1	[-1, 1]	3				Exp Growth
Tanh	219	219	73	0.0005	0.0001	2	0.25	0.9	500	FALSE	1	0	5	Uniform	0.05	0.5	1	FALSE	ZO	[0, 1]	2	0.1929	0.1451	221.7	
Tanh	73	219	219	0.0005	0	0	0.25	0.999	500	FALSE	1	1	5	Normal	0.5	0.5	1	TRUE	PM.5	[-0.5, 0.5]	3	0.5100	0.4898	174.9	
Tanh	73	219	219	0.05	0.0001	0	0.25	0.999	500	TRUE	0	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	PM1	[-1, 1]	2	0.8728	0.9731	166.7	
Tanh	219	219	219	0.0005	0	2	0.25	0.999	50000	FALSE	0	1	500	Uniform	0.05	2	10000	TRUE	PM.5	[-0.5, 0.5]	2	0.1588	0.1090	152.3	
Rectifier	219	219	219	0.05	0	2	0.25	0.999	50000	TRUE	1	0	5	Uniform Adaptive	0.5	2	1	FALSE	PM.5	[-0.5, 0.5]	1				Exp Growth
Tanh	73	73	219	0.0005	0	0	0.25	0.9	500	TRUE	0	0	500	Uniform	0.05	2	1	FALSE	PM1	[-1, 1]	1	1.6169	1.5385	114.7	
Tanh	73	73	73	0.05	0	2	0.75	0.999	500	FALSE	0	0	500	Uniform Adaptive	0.5	0.5	1	FALSE	PM.5	[-0.5, 0.5]	2				Exp Growth
Tanh	219	73	219	0.05	0	2	0.75	0.9	500	TRUE	1	1	500	Normal	0.05	0.5	1	TRUE	PM.5	[-0.5, 0.5]	2	0.7833	0.6322	162.4	
Rectifier	219	73	73	0.0005	0.0001	2	0.25	0.999	50000	TRUE	0	1	5	Uniform Adaptive	0.5	0.5	1	TRUE	ZO	[0, 1]	2				Exp Growth
Rectifier	219	73	219	0.0005	0.0001	0	0.75	0.999	500	FALSE	0	0	500	Uniform	0.5	0.5	1	TRUE	PM1	[-1, 1]	1				Exp Growth
Rectifier	73	219	219	0.05	0.0001	2	0.75	0.999	500	TRUE	1	1	500	Normal	0.05	0.5	10000	TRUE	ZO	[0, 1]	1				Exp Growth
Rectifier	73	73	219	0.0005	0	0	0.75	0.9	50000	TRUE	0	0	5	Normal	0.5	0.5	10000	FALSE	PM.5	[-0.5, 0.5]	2	13.9523	14.6947	86.0	
Tanh	73	219	73	0.05	0	2	0.25	0.9	50000	FALSE	0	0	500	Normal	0.5	0.5	1	FALSE	ZO	[0, 1]	1	0.9935	0.9505	67.7	
Tanh	73	73	73	0.05	0.0001	0	0.75	0.999	50000	FALSE	1	1	5	Uniform	0.05	2	10000	FALSE	PM.5	[-0.5, 0.5]	1	0.4810	0.4975	153.8	
Rectifier	73	219	73	0.0005	0.0001	0	0.25	0.999	50000	FALSE	1	1	500	Normal	0.5	2	1	FALSE	PM1	[-1, 1]	2				Exp Growth
Rectifier	219	73	219	0.05	0	2	0.25	0.9	500	FALSE	1	1	5	Uniform	0.5	0.5	10000	FALSE	PM1	[-1, 1]	3				Exp Growth
Rectifier	219	219	219	0.05	0	0	0.75	0.9	50000	FALSE	1	0	500	Uniform Adaptive	0.05	2	10000	FALSE	ZO	[0, 1]	2	0.6663	0.6738	208.5	
Tanh	219	219	219	0.05	0.0001	0	0.75	0.999	500	TRUE	0	1	500	Uniform	0.5	2	1	FALSE	ZO	[0, 1]	3	1.0596	1.0204	192.9	
Rectifier	219	219	73	0.0005	0.0001	0	0.75	0.9	500	TRUE	1	1	500	Uniform Adaptive	0.5	0.5	10000	FALSE	PM.5	[-0.5, 0.5]	3	0.5662	0.5834	98.9	
Tanh	219	219	73	0.05	0	0	0.75	0.9	50000	FALSE	0	1	5	Uniform Adaptive	0.05	0.5	1	TRUE	PM1	[-1, 1]	1	0.7409	0.8497	91.2	
Tanh	219	73	73	0.0005	0	2	0.75	0.999	50000	TRUE	0	0	500	Normal	0.05	0.5	10000	FALSE	PM1	[-1, 1]	3	0.6943	0.4295	62.7	
Rectifier	73	219	73	0.05	0	2	0.75	0.9	500	TRUE	0	1	5	Uniform	0.5	2	10000	TRUE	PM1	[-1, 1]	2				Exp Growth
Rectifier	73	73	219	0.0005	0.0001	2	0.75	0.9	50000	FALSE	0	1	5	Uniform Adaptive	0.05	2	1	FALSE	ZO	[0, 1]	3				Exp Growth
Tanh	146	73	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	PM.5	[-0.5, 0.5]	1	0.1380	0.1322	88.9	
Rectifier	146	73	146	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	PM1	[-1, 1]	2				Exp Growth
Tanh	73	146	73	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	ZO	[0, 1]	3	0.1560	0.1047	43.9	
Rectifier	73	146	73	0.02525	0.00005	1	0.5	0.9495	10000	TRUE	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	PM.5	[-0.5, 0.5]	1				Exp Growth
Tanh	146	146	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	PM1	[-1, 1]	2	0.5192	0.4088	143.8	
Rectifier	146	146	146	0.02525	0.00005	1	0.5	0.9495	10000	FALSE	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	PM1	[-1, 1]	2				Exp Growth

Appendix N: Screening D_ME1DOAR

Table 31. Screening D_ME1DOAR Test Matrix

Activation Function	Neurons Per Hidden Layer	Rho	Epsilon	Nesterov Accelerated Gradient	Input Dropout Rate	Hidden Dropout Rate	L1	L2	Max W2	Initial Weight Distribution	Average Activation	Sparsity Beta	Minibatch Size	Shuffle Training Data	Data Scale	Test/Training Set	Center	Test MSE	Train MSE	Time (Sec)	Notes
RectifierWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	0	0	500	Normal	0.5	0.5	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	128.9	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.2	0.2	1	0	5	Uniform Adaptive	0.05	2	10000	TRUE	[-0.5, 05]	3	0	0.1968	0.2448	114.4	
TanhWithDropout	219	0.999	1E-10	TRUE	0.2	0.8	1	0	500	Uniform Adaptive	0.5	2	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2469	155.9	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.2	1	0	5	Normal	0.5	0.5	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	115.9	
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	0	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 05]	1	0	0.1215	0.0748	30.9	
TanhWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	0	1	5	Normal	0.5	2	10000	TRUE	[-0.5, 05]	2	0	0.1152	0.0711	136.2	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.2	0.8	1	1	5	Uniform	0.05	0.5	10000	FALSE	[-0.5, 05]	3	0				Exp Growth
TanhWithDropout	73	0.999	0.000001	FALSE	0.8	0.2	0	1	500	Uniform	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1320	0.0910	69.8	
RectifierWithDropout	146	0.9495	1E-10	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1999	0.2427	89.8	
TanhWithDropout	146	0.9495	1E-10	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1416	0.1386	174.1	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	0	1	5	Uniform	0.5	2	1	TRUE	[0, 1]	2	0	0.1093	0.0566	50.1	
TanhWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	1	1	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	1	0	0.1927	0.1465	263.5	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.8	0	0	5	Uniform	0.05	0.5	1	TRUE	[0, 1]	1	0	0.0979	0.0436	117.0	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.8	0	0	5	Normal	0.05	2	10000	FALSE	[0, 1]	1	0				Exp Growth
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.2	0	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	3	0	0.2230	0.1950	56.8	
TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	1	0	500	Normal	0.5	0.5	10000	TRUE	[0, 1]	3	0	0.1657	0.1173	34.8	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	0	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	2	0	0.0818	0.0315	58.9	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.2	1	0	500	Normal	0.05	2	1	FALSE	[0, 1]	3	0	0.1944	0.1469	270.5	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.8	0.8	1	1	500	Uniform	0.5	0.5	10000	FALSE	[0, 1]	1	0	0.1790	0.1474	26.3	
RectifierWithDropout	146	0.9495	1E-10	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1741	0.0753	67.4	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	0	0	500	Normal	0.5	0.5	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	118.1	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.2	0.2	1	0	5	Uniform Adaptive	0.05	2	10000	TRUE	[-0.5, 05]	3	0	0.1969	0.2448	87.9	
TanhWithDropout	219	0.999	1E-10	TRUE	0.2	0.8	1	0	500	Uniform Adaptive	0.5	2	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2469	151.5	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.2	1	0	5	Normal	0.5	0.5	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	92.8	
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	0	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 05]	1	0	0.1206	0.0753	48.9	
TanhWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	0	1	5	Normal	0.5	2	10000	TRUE	[-0.5, 05]	2	0	0.1065	0.0753	94.3	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.2	0.8	1	1	5	Uniform	0.05	0.5	10000	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	80.0	
TanhWithDropout	73	0.999	0.000001	FALSE	0.8	0.2	0	1	500	Uniform	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1376	0.0969	74.1	
RectifierWithDropout	146	0.9495	1E-10	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1996	0.2425	103.6	
TanhWithDropout	146	0.9495	1E-10	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1438	0.1425	88.3	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	0	1	5	Uniform	0.5	2	1	TRUE	[0, 1]	2	0	0.1095	0.0580	47.6	
TanhWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	1	1	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	1	0	0.1757	0.1247	112.9	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.8	0	0	5	Uniform	0.05	0.5	1	TRUE	[0, 1]	1	0	0.0973	0.0433	114.3	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.8	0	0	5	Normal	0.05	2	10000	FALSE	[0, 1]	1	0				Exp Growth
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.2	0	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	3	0	0.2323	0.2114	27.0	
TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	1	0	500	Normal	0.5	0.5	10000	TRUE	[0, 1]	3	0	0.1618	0.1118	27.6	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	0	0	5	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	2	0	0.0820	0.0319	58.5	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.2	1	0	500	Normal	0.05	2	1	FALSE	[0, 1]	3	0	0.1945	0.1471	250.2	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.8	0.8	1	1	500	Uniform	0.5	0.5	10000	FALSE	[0, 1]	1	0	0.1489	0.0942	45.0	
RectifierWithDropout	146	0.9495	1E-10	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.2859	0.1194	66.8	
TanhWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	0	0	5	Uniform Adaptive	0.5	0.5	1	FALSE	[-0.5, 05]	3	0	0.1327	0.0781	69.0	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.8	0.2	1	0	500	Uniform	0.05	0.5	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	112.2	
TanhWithDropout	219	0.999	1E-10	FALSE	0.8	0.2	1	1	5	Normal	0.05	0.5	10000	TRUE	[0, 1]	2	0	0.1950	0.1472	121.4	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.2	0.2	0	1	500	Normal	0.05	2	10000	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	55.8	
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.2	1	0	5	Uniform	0.5	2	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2469	46.8	

RectifierWithDropout	73	0.999	0.000001	FALSE	0.8	0.8	1	0	500	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	2	0	0.1849	0.1447	10.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.2	0.8	1	1	500	Normal	0.05	0.5	1	FALSE	[-0.5, 05]	2	0	0.1962	0.2428	119.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.2	0.2	0	0	500	Uniform	0.5	2	10000	TRUE	[0, 1]	1	0	0.1683	0.1318	193.8	
TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	1	1	500	Uniform	0.05	0.5	1	TRUE	[-0.5, 05]	2	0	0.1723	0.2036	68.4	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	1	0	5	Uniform	0.05	0.5	1	TRUE	[0, 1]	3	0	0.1891	0.1419	76.6	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	0	1	5	Normal	0.5	0.5	1	FALSE	[0, 1]	3	0	0.1000	0.0440	27.9	
TanhWithDropout	73	0.999	1E-10	TRUE	0.2	0.2	1	1	5	Normal	0.5	0.5	1	TRUE	[0, 1]	3	0	0.1965	0.1491	47.6	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	1	0	500	Normal	0.5	2	10000	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	301.7	
TanhWithDropout	219	0.9	0.000001	FALSE	0.8	0.2	1	1	500	Uniform	0.5	2	1	TRUE	[0, 1]	2	0	0.1922	0.1433	116.5	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.8	0.8	0	1	500	Uniform	0.5	0.5	10000	TRUE	[0, 1]	2	0	0.1726	0.0844	72.4	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	0	1	5	Uniform Adaptive	0.05	0.5	1	TRUE	[0, 1]	3	0	0.1082	0.0530	55.6	
TanhWithDropout	219	0.999	1E-10	TRUE	0.8	0.2	1	1	500	Normal	0.5	2	1	TRUE	[0, 1]	2	0	0.1947	0.1468	116.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.2	0.8	0	1	500	Uniform Adaptive	0.05	2	10000	FALSE	[-0.5, 05]	3	0	0.1041	0.0673	124.4	
TanhWithDropout	73	0.999	1E-10	FALSE	0.8	0.2	0	0	5	Normal	0.5	0.5	10000	FALSE	[0, 1]	1	0	0.1082	0.0545	82.8	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	1	1	5	Uniform	0.05	2	10000	TRUE	[-0.5, 05]	2	0	0.1964	0.2426	78.6	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.8	0.2	0	0	5	Uniform Adaptive	0.05	0.5	1	FALSE	[-0.5, 05]	3	0	0.1971	0.2451	58.3	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.8	0.2	0	0	500	Uniform Adaptive	0.5	2	10000	TRUE	[-0.5, 05]	2	0	0.4641	0.5658	104.1	
TanhWithDropout	73	0.9	1E-10	FALSE	0.8	0.2	1	0	5	Uniform Adaptive	0.05	0.5	1	TRUE	[-0.5, 05]	2	0	0.1990	0.2469	59.2	
RectifierWithDropout	219	0.9	0.000001	TRUE	0.2	0.8	1	0	5	Uniform	0.5	0.5	1	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	89.3	
TanhWithDropout	219	0.9	1E-10	FALSE	0.8	0.8	0	0	500	Uniform Adaptive	0.05	2	10000	TRUE	[0, 1]	3	0	0.1051	0.0501	291.1	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.8	0.8	0	1	500	Normal	0.05	0.5	1	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	41.9	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.2	1	0	500	Uniform	0.5	0.5	10000	FALSE	[0, 1]	2	0	0.9769	0.9561	190.4	
RectifierWithDropout	219	0.9	1E-10	TRUE	0.8	0.8	1	1	500	Normal	0.05	2	10000	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	80.3	
TanhWithDropout	219	0.999	1E-10	TRUE	0.2	0.2	0	1	500	Uniform	0.05	0.5	1	TRUE	[0, 1]	2	0	0.1071	0.0408	121.5	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	1	0	5	Uniform	0.5	2	10000	FALSE	[0, 1]	1	0				Exp Growth
TanhWithDropout	73	0.999	0.000001	TRUE	0.8	0.8	1	1	500	Normal	0.5	2	10000	FALSE	[-0.5, 05]	3	0	0.1649	0.1837	41.4	
TanhWithDropout	73	0.9	1E-10	FALSE	0.2	0.8	1	1	500	Uniform Adaptive	0.05	2	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2469	52.3	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.2	1	0	5	Uniform	0.5	0.5	1	FALSE	[0, 1]	1	0	0.1919	0.1493	119.1	
TanhWithDropout	219	0.999	1E-10	FALSE	0.8	0.8	0	0	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	2	0	0.1406	0.0902	66.0	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	1	1	5	Normal	0.5	2	1	FALSE	[-0.5, 05]	3	0	0.1967	0.2447	49.0	
TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	1	1	5	Uniform Adaptive	0.5	2	1	FALSE	[-0.5, 05]	2	0	0.1366	0.1402	71.5	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.2	1	1	5	Uniform	0.05	0.5	10000	FALSE	[0, 1]	1	0	0.1372	0.0880	13.8	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.8	0	0	500	Uniform Adaptive	0.05	0.5	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	64.5	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.2	0.2	0	0	5	Uniform	0.5	0.5	10000	TRUE	[-0.5, 05]	3	0	0.1951	1.0528	209.3	
TanhWithDropout	219	0.9	1E-10	FALSE	0.2	0.8	1	1	5	Normal	0.05	2	1	TRUE	[0, 1]	3	0	0.1963	0.1488	97.6	
TanhWithDropout	219	0.999	1E-10	FALSE	0.8	0.2	1	1	500	Uniform	0.5	0.5	1	FALSE	[-0.5, 05]	1	0	0.1997	0.2422	110.4	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	0	0	500	Uniform Adaptive	0.5	2	10000	TRUE	[0, 1]	1	0	0.1008	0.0623	50.5	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	0	0	500	Uniform	0.5	2	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	133.9	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.8	1	1	5	Normal	0.5	2	10000	FALSE	[0, 1]	2	0	0.1884	0.1408	33.3	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.2	0.2	0	0	500	Uniform Adaptive	0.05	2	1	FALSE	[0, 1]	3	0	0.0696	0.0194	24.9	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.2	0.2	0	1	500	Uniform	0.5	0.5	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	61.9	
RectifierWithDropout	73	0.999	1E-10	FALSE	0.8	0.8	0	0	500	Normal	0.5	2	10000	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	20.7	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.8	1	1	5	Uniform	0.5	2	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2469	104.0	
TanhWithDropout	73	0.9	1E-10	TRUE	0.8	0.2	0	1	5	Uniform Adaptive	0.5	2	1	FALSE	[-0.5, 05]	1	0	0.1670	0.0937	49.5	
RectifierWithDropout	73	0.999	1E-10	FALSE	0.2	0.8	1	1	500	Uniform	0.5	2	10000	TRUE	[-0.5, 05]	3	0	0.1972	0.2452	18.5	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.8	0.8	0	1	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	3	0				Exp Growth
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.8	1	1	500	Normal	0.05	0.5	10000	FALSE	[0, 1]	1	0	0.1798	0.1353	78.9	
TanhWithDropout	73	0.9	1E-10	FALSE	0.8	0.2	0	0	5	Uniform	0.05	2	1	FALSE	[0, 1]	2	0	0.1018	0.0502	157.2	
TanhWithDropout	73	0.9	0.000001	TRUE	0.8	0.2	0	1	5	Normal	0.05	0.5	10000	FALSE	[-0.5, 05]	3	0	0.1283	0.0730	50.0	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.2	0	1	5	Uniform Adaptive	0.5	2	10000	FALSE	[0, 1]	2	0	0.0919	0.0392	93.1	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.8	0	1	500	Uniform Adaptive	0.05	2	10000	FALSE	[0, 1]	2	0	0.1862	0.1446	27.4	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.8	0.8	1	0	5	Uniform Adaptive	0.5	2	1	TRUE	[0, 1]	3	0	0.1840	0.1439	51.6	
TanhWithDropout	219	0.999	1E-10	TRUE	0.8	0.8	0	0	5	Uniform	0.05	0.5	1	FALSE	[-0.5, 05]	2	0	0.0916	0.0427	176.1	
TanhWithDropout	73	0.999	1E-10	FALSE	0.8	0.8	1	1	5	Uniform Adaptive	0.5	2	10000	TRUE	[0, 1]	1	0	0.1934	0.1508	28.4	
TanhWithDropout	219	0.999	0.000001	FALSE	0.8	0.2	0	0	500	Uniform Adaptive	0.05	2	10000	FALSE	[-0.5, 05]	2	0	0.2866	0.3573	149.3	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.2	0.8	1	0	500	Uniform	0.05	2	10000	FALSE	[0, 1]	3	0	0.1515	0.0958	20.6	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	0	1	5	Uniform Adaptive	0.05	0.5	1	FALSE	[-0.5, 05]	3	0	0.0949	0.0565	207.3	
TanhWithDropout	73	0.9	1E-10	FALSE	0.8	0.8	1	0	500	Normal	0.5	0.5	1	FALSE	[0, 1]	2	0	0.1584	0.1081	33.4	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.2	0.2	1	0	5	Normal	0.05	0.5	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	36.0	
TanhWithDropout	73	0.9	1E-10	TRUE	0.2	0.2	0	0	5	Normal	0.5	0.5	1	FALSE	[-0.5, 05]	2	0	0.0927	0.0485	96.2	
TanhWithDropout	73	0.999	0.000001	TRUE	0.2	0.8	0	1	500	Uniform Adaptive	0.5	0.5	10000	FALSE	[0, 1]	1	0	0.1294	0.0769	79.0	
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.2	0	0	500	Uniform	0.5	0.5	1	TRUE	[0, 1]	3	0	0.1102	0.0571	192.4	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.8	0.2	0	0	500	Normal	0.5	2	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	46.5	
TanhWithDropout	73	0.999	0.000001	TRUE	0.2	0.8	0	1	500	Normal	0.5	2	1	TRUE	[0, 1]	1	0	0.1249	0.0726	55.0	

TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.8	0	1	5	Normal	0.05	2	1	FALSE	[0, 1]	2	0	0.1241	0.0729	45.7	
TanhWithDropout	73	0.999	1E-10	FALSE	0.2	0.8	0	0	500	Uniform Adaptive	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1116	0.0572	45.4	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.8	1	0	5	Normal	0.5	0.5	1	FALSE	[-0.5, 05]	1	0	0.1406	0.1415	131.7	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.2	0.2	0	1	500	Uniform	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	116.0	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.8	0.2	1	1	500	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	115.8	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	0	0	5	Normal	0.5	2	1	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	145.8	
TanhWithDropout	219	0.999	1E-10	FALSE	0.8	0.8	1	1	5	Uniform	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	92.5	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.2	0.8	1	0	5	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	107.4	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.8	0.2	1	1	500	Uniform	0.5	0.5	1	TRUE	[0, 1]	1	0	0.2512	0.3100	13.7	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	0	1	500	Normal	0.05	0.5	10000	TRUE	[-0.5, 05]	3	0	0.0991	0.0569	184.6	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.2	0.8	1	0	500	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	2	0	0.1487	0.0989	22.1	
TanhWithDropout	73	0.9	0.000001	FALSE	0.8	0.8	0	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[-0.5, 05]	3	0	0.1140	0.0768	47.9	
RectifierWithDropout	219	0.9	1E-10	TRUE	0.8	0.8	0	0	500	Uniform	0.5	2	1	FALSE	[0, 1]	2	0	0.2039	0.1059	95.9	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.2	0	1	5	Uniform	0.5	2	1	FALSE	[-0.5, 05]	2	0	0.1963	0.1996	215.8	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	0	1	500	Normal	0.05	2	10000	TRUE	[0, 1]	2	0	0.1598	0.0690	61.1	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.2	0	0	500	Uniform Adaptive	0.05	2	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	43.0	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.2	0.2	0	0	500	Normal	0.5	0.5	10000	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	88.3	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.8	0	0	500	Normal	0.05	0.5	1	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	40.5	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.8	0.8	0	0	5	Normal	0.05	0.5	1	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	57.2	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	1	0	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	1	0	0.1371	0.0862	87.3	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.8	0.2	0	1	5	Uniform	0.05	0.5	1	FALSE	[0, 1]	2	0				Exp Growth
TanhWithDropout	219	0.9	1E-10	FALSE	0.2	0.8	0	1	500	Uniform Adaptive	0.5	2	1	TRUE	[-0.5, 05]	2	0	0.1081	0.0640	108.8	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.8	0.2	0	1	500	Normal	0.5	2	10000	TRUE	[0, 1]	3	0	4.7390	0.7351	48.4	
TanhWithDropout	219	0.9	1E-10	TRUE	0.2	0.8	1	1	5	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	2	0	0.1950	0.1472	97.7	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.2	0.2	1	0	500	Uniform	0.5	2	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	37.1	
TanhWithDropout	219	0.9	1E-10	TRUE	0.2	0.8	1	1	500	Uniform	0.5	2	10000	FALSE	[0, 1]	3	0	0.1519	0.1033	65.8	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.2	0.2	1	1	5	Uniform Adaptive	0.5	2	1	FALSE	[0, 1]	2	0	0.1374	0.0872	38.8	
RectifierWithDropout	73	0.9	1E-10	FALSE	0.8	0.8	0	0	5	Uniform Adaptive	0.5	0.5	10000	FALSE	[-0.5, 05]	2	0	0.1987	0.2466	34.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.8	0.2	1	1	5	Uniform	0.05	0.5	1	TRUE	[-0.5, 05]	1	0	0.1969	0.2389	153.4	
TanhWithDropout	73	0.9	1E-10	TRUE	0.8	0.2	1	0	5	Normal	0.05	2	10000	TRUE	[0, 1]	1	0	0.1919	0.1493	54.9	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.8	0.8	0	1	5	Normal	0.5	0.5	10000	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	55.8	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.8	0.2	1	1	5	Normal	0.05	0.5	1	FALSE	[-0.5, 05]	1	0	0.1997	0.2427	111.3	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.2	0.8	1	1	500	Normal	0.5	2	1	FALSE	[0, 1]	2	0	0.1694	0.1151	35.3	
RectifierWithDropout	73	0.9	1E-10	TRUE	0.8	0.2	1	1	500	Uniform	0.05	0.5	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	45.3	
TanhWithDropout	73	0.9	1E-10	FALSE	0.2	0.8	0	1	5	Normal	0.5	0.5	10000	FALSE	[-0.5, 05]	1	0	0.1322	0.0847	76.2	
RectifierWithDropout	219	0.9	1E-10	TRUE	0.2	0.2	1	1	500	Normal	0.5	0.5	1	TRUE	[0, 1]	2	0	0.7250	0.0843	108.9	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.2	0.2	1	0	5	Uniform Adaptive	0.5	2	1	TRUE	[0, 1]	2	0				Exp Growth
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	1	1	500	Uniform	0.05	2	1	FALSE	[0, 1]	1	0	0.1935	0.1510	31.2	
TanhWithDropout	73	0.9	1E-10	FALSE	0.2	0.8	0	1	500	Uniform	0.05	2	10000	TRUE	[-0.5, 05]	1	0	0.1373	0.0896	69.2	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	1	1	500	Uniform Adaptive	0.05	2	1	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	136.3	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.8	0.2	1	1	5	Uniform	0.5	0.5	10000	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	34.8	
TanhWithDropout	73	0.999	1E-10	FALSE	0.2	0.2	1	0	500	Normal	0.5	2	1	TRUE	[-0.5, 05]	2	0	0.1951	0.2408	61.9	
TanhWithDropout	73	0.999	0.000001	TRUE	0.8	0.2	1	0	500	Uniform Adaptive	0.05	2	10000	FALSE	[0, 1]	1	0				Exp Growth
RectifierWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	0	1	5	Uniform Adaptive	0.5	2	10000	FALSE	[0, 1]	3	0	0.1001	0.0449	41.6	
TanhWithDropout	73	0.9	0.000001	TRUE	0.8	0.8	1	0	500	Uniform Adaptive	0.5	2	1	TRUE	[-0.5, 05]	2	0	0.1877	0.2294	31.8	
RectifierWithDropout	73	0.999	1E-10	FALSE	0.2	0.8	0	0	5	Uniform	0.5	0.5	1	FALSE	[0, 1]	3	0	0.1147	0.0608	46.1	
TanhWithDropout	73	0.999	1E-10	TRUE	0.2	0.2	0	0	5	Normal	0.05	2	1	TRUE	[-0.5, 05]	1	0	0.0785	0.0354	211.8	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	1	0	5	Uniform	0.05	0.5	10000	TRUE	[-0.5, 05]	1	0	0.1998	0.2426	27.3	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.8	0.2	0	0	500	Normal	0.05	0.5	1	TRUE	[0, 1]	2	0	0.8313	0.2832	11.7	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.8	1	1	500	Uniform	0.5	2	10000	TRUE	[-0.5, 05]	1	0	0.1597	0.1563	184.8	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.8	0.2	1	0	500	Uniform	0.5	2	10000	FALSE	[0, 1]	3	0	0.6835	0.3396	36.0	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.8	0	0	500	Uniform Adaptive	0.05	0.5	1	FALSE	[0, 1]	3	0	0.2073	0.1572	92.3	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.8	0	1	5	Uniform	0.5	0.5	1	FALSE	[-0.5, 05]	1	0	0.1997	0.2425	94.6	
TanhWithDropout	219	0.999	1E-10	TRUE	0.8	0.2	0	0	5	Uniform	0.05	2	10000	FALSE	[-0.5, 05]	1	0	0.0941	0.0639	119.9	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.8	0.2	1	1	5	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	1	0	0.1940	0.1514	35.5	
TanhWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	0	1	500	Normal	0.5	0.5	1	FALSE	[-0.5, 05]	3	0	0.1372	0.0898	44.6	
TanhWithDropout	73	0.9	0.000001	FALSE	0.8	0.8	1	0	5	Uniform Adaptive	0.05	0.5	1	FALSE	[0, 1]	1	0	0.1822	0.1396	45.2	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.8	0	0	5	Uniform	0.5	2	10000	TRUE	[-0.5, 05]	2	0	0.1250	0.0742	93.6	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.2	0.8	1	0	5	Normal	0.05	2	1	TRUE	[0, 1]	1	0	0.1716	0.1288	13.5	
TanhWithDropout	73	0.9	1E-10	TRUE	0.2	0.2	0	0	500	Uniform Adaptive	0.5	0.5	10000	FALSE	[0, 1]	3	0	0.0951	0.0410	56.6	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.8	0.2	1	1	500	Normal	0.05	2	1	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	45.6	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.2	0	1	5	Normal	0.05	0.5	10000	TRUE	[0, 1]	1	0	0.1241	0.0689	244.2	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.2	0.2	0	1	500	Normal	0.05	2	10000	FALSE	[-0.5, 05]	2	0	0.3926	0.2464	65.6	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	0	0	5	Normal	0.5	0.5	1	TRUE	[0, 1]	2	0	0.0888	0.0449	27.6	
TanhWithDropout	219	0.999	0.000001	TRUE	0.2	0.2	0	0	500	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 05]	1	0	0.2665	0.3146	186.5	

RectifierWithDropout	219	0.9	0.000001	TRUE	0.8	0.8	0	1	500	Normal	0.5	0.5	10000	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	55.5	
TanhWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	0	1	5	Uniform	0.05	0.5	10000	FALSE	[-0.5, 05]	2	0	0.1019	0.0669	126.3	
TanhWithDropout	73	0.999	1E-10	FALSE	0.8	0.8	1	0	5	Normal	0.05	0.5	10000	TRUE	[-0.5, 05]	3	0	0.1972	0.2451	32.3	
RectifierWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	1	1	5	Uniform	0.5	0.5	1	TRUE	[0, 1]	3	0				Exp Growth
TanhWithDropout	219	0.999	0.000001	TRUE	0.8	0.8	1	0	5	Uniform	0.05	2	10000	TRUE	[0, 1]	2	0	0.1589	0.1114	103.9	
TanhWithDropout	219	0.999	0.000001	TRUE	0.8	0.2	1	0	5	Uniform Adaptive	0.5	2	10000	TRUE	[-0.5, 05]	3	0				Exp Growth
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.8	0	1	5	Uniform	0.5	2	10000	FALSE	[-0.5, 05]	1	0	0.1997	0.2425	48.8	
TanhWithDropout	219	0.999	0.000001	FALSE	0.8	0.8	0	1	5	Uniform	0.5	0.5	10000	FALSE	[0, 1]	3	0	0.1281	0.0732	137.7	
TanhWithDropout	73	0.999	0.000001	FALSE	0.2	0.2	0	1	500	Uniform	0.5	2	10000	FALSE	[0, 1]	2	0	0.1199	0.0597	64.1	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.2	0.8	1	1	500	Uniform Adaptive	0.5	0.5	1	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	28.4	
TanhWithDropout	73	0.9	0.000001	TRUE	0.8	0.8	0	0	5	Normal	0.5	2	1	FALSE	[0, 1]	3	0	0.1446	0.0901	48.6	
TanhWithDropout	73	0.9	1E-10	FALSE	0.8	0.2	1	1	500	Uniform	0.05	0.5	10000	TRUE	[0, 1]	3	0	0.1965	0.1491	58.6	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.8	0	1	5	Uniform	0.5	0.5	1	TRUE	[-0.5, 05]	3	0	0.1195	0.0759	63.8	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	1	0	5	Normal	0.05	2	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	17.8	
RectifierWithDropout	219	0.9	1E-10	TRUE	0.2	0.8	0	0	5	Normal	0.05	0.5	10000	FALSE	[0, 1]	2	0	0.0872	0.0346	98.5	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.8	0.2	1	0	5	Normal	0.5	0.5	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	59.9	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.8	0	0	500	Normal	0.5	0.5	10000	TRUE	[-0.5, 05]	1	0	0.1380	0.0987	162.9	
TanhWithDropout	219	0.999	0.000001	FALSE	0.2	0.2	1	0	500	Uniform	0.05	2	10000	FALSE	[-0.5, 05]	3	0	0.6182	0.5455	175.4	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.2	0.8	1	0	5	Uniform	0.05	2	1	FALSE	[0, 1]	2	0	0.1486	0.0965	63.8	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.8	0.2	1	1	500	Normal	0.5	0.5	10000	FALSE	[0, 1]	1	0	0.9906	0.3397	51.8	
RectifierWithDropout	73	0.999	1E-10	FALSE	0.2	0.2	0	1	500	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	1	0	0.1010	0.0476	25.5	
RectifierWithDropout	73	0.9	0.000001	TRUE	0.8	0.8	1	0	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	3	0	0.1774	0.1298	18.4	
TanhWithDropout	219	0.9	0.000001	TRUE	0.2	0.8	1	0	5	Uniform Adaptive	0.05	2	1	TRUE	[-0.5, 05]	1	0	0.1976	0.2392	136.6	
TanhWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	1	0	500	Uniform Adaptive	0.05	0.5	10000	TRUE	[0, 1]	2	0	0.1832	0.1353	99.1	
TanhWithDropout	219	0.9	1E-10	TRUE	0.8	0.2	0	1	5	Uniform	0.05	2	1	TRUE	[0, 1]	3	0	0.1363	0.0826	374.6	
TanhWithDropout	219	0.9	1E-10	FALSE	0.8	0.2	0	1	500	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	1	0	0.1360	0.0762	223.3	
RectifierWithDropout	219	0.9	0.000001	TRUE	0.8	0.2	1	1	5	Normal	0.05	2	1	FALSE	[0, 1]	2	0	0.1821	0.1343	39.2	
TanhWithDropout	73	0.9	0.000001	FALSE	0.2	0.2	1	0	500	Uniform Adaptive	0.5	0.5	1	FALSE	[-0.5, 05]	3	0	0.1965	0.2441	66.6	
TanhWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	0	0	500	Normal	0.5	2	10000	FALSE	[0, 1]	1	0	0.0978	0.0482	268.9	
RectifierWithDropout	73	0.999	1E-10	FALSE	0.8	0.2	0	1	5	Uniform	0.5	2	1	TRUE	[-0.5, 05]	2	0	0.1990	0.2470	44.4	
RectifierWithDropout	219	0.999	1E-10	FALSE	0.2	0.2	1	1	5	Normal	0.5	2	10000	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	164.2	
TanhWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	1	0	500	Uniform Adaptive	0.5	2	10000	FALSE	[0, 1]	1	0	0.1695	0.1259	107.2	
TanhWithDropout	73	0.9	0.000001	TRUE	0.2	0.8	1	0	500	Uniform	0.5	0.5	10000	FALSE	[-0.5, 05]	1	0	0.1983	0.2402	49.0	
TanhWithDropout	73	0.9	1E-10	TRUE	0.2	0.2	1	1	500	Uniform Adaptive	0.05	2	1	TRUE	[0, 1]	3	0	0.1965	0.1491	66.7	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.8	0.2	1	0	5	Uniform	0.05	2	1	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	55.1	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.2	0	1	500	Normal	0.5	2	1	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	120.1	
TanhWithDropout	73	0.999	1E-10	TRUE	0.2	0.8	0	0	500	Normal	0.05	0.5	10000	TRUE	[0, 1]	2	0	0.1093	0.0581	123.7	
RectifierWithDropout	219	0.9	1E-10	FALSE	0.8	0.8	0	1	5	Normal	0.5	2	1	FALSE	[0, 1]	1	0	0.1250	0.0723	33.1	
TanhWithDropout	73	0.9	0.000001	FALSE	0.2	0.2	0	0	5	Uniform Adaptive	0.5	2	10000	TRUE	[-0.5, 05]	1	0	0.1016	0.0540	69.6	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.8	0.8	0	1	500	Normal	0.05	0.5	10000	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	23.0	
TanhWithDropout	73	0.999	1E-10	FALSE	0.2	0.2	0	1	5	Uniform	0.5	2	10000	TRUE	[0, 1]	3	0	0.1031	0.0483	74.4	
RectifierWithDropout	219	0.9	0.000001	TRUE	0.2	0.8	1	1	500	Uniform	0.05	0.5	1	FALSE	[0, 1]	1	0	0.1662	0.1216	58.5	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.8	0.8	1	0	500	Uniform Adaptive	0.05	0.5	10000	FALSE	[0, 1]	3	0	0.1823	0.1456	51.5	
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.8	1	0	500	Uniform	0.5	0.5	1	FALSE	[-0.5, 05]	3	0	0.1973	0.2452	55.8	
TanhWithDropout	219	0.9	1E-10	FALSE	0.8	0.2	1	1	500	Normal	0.5	2	10000	FALSE	[-0.5, 05]	3	0	0.1971	0.2448	123.5	
TanhWithDropout	73	0.999	0.000001	TRUE	0.8	0.8	0	0	5	Uniform	0.5	2	1	TRUE	[-0.5, 05]	1	0	0.1613	0.1070	30.3	
TanhWithDropout	219	0.999	0.000001	TRUE	0.8	0.8	1	1	5	Uniform Adaptive	0.5	0.5	10000	FALSE	[-0.5, 05]	1	0	0.1424	0.1409	104.5	
TanhWithDropout	219	0.999	0.000001	FALSE	0.8	0.2	1	0	500	Normal	0.05	2	1	TRUE	[0, 1]	1	0	0.9349	0.9493	244.7	
TanhWithDropout	219	0.9	0.000001	FALSE	0.2	0.2	1	0	5	Normal	0.05	0.5	10000	FALSE	[0, 1]	3	0	0.1894	0.1421	188.1	
TanhWithDropout	73	0.9	0.000001	FALSE	0.2	0.2	0	1	500	Uniform Adaptive	0.05	0.5	1	TRUE	[0, 1]	2	0	0.1052	0.0531	52.1	
TanhWithDropout	73	0.999	1E-10	TRUE	0.2	0.2	1	1	500	Uniform Adaptive	0.5	0.5	10000	TRUE	[-0.5, 05]	2	0	0.1990	0.2469	85.5	
TanhWithDropout	73	0.999	1E-10	FALSE	0.8	0.8	1	0	500	Uniform	0.05	0.5	10000	FALSE	[0, 1]	2	0	0.1926	0.1448	29.2	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.2	1	1	500	Uniform Adaptive	0.5	2	1	FALSE	[-0.5, 05]	3	0	0.1969	0.2452	222.3	
RectifierWithDropout	73	0.9	0.000001	FALSE	0.8	0.8	0	0	500	Uniform	0.5	2	1	TRUE	[0, 1]	3	0	0.1400	0.0766	12.1	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.8	0	0	500	Normal	0.05	2	1	TRUE	[-0.5, 05]	3	0	0.1973	0.2452	88.8	
TanhWithDropout	73	0.999	1E-10	FALSE	0.8	0.2	1	1	500	Uniform Adaptive	0.5	0.5	1	FALSE	[0, 1]	3	0	0.1962	0.1487	49.7	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.2	0.8	0	1	500	Uniform	0.05	0.5	10000	TRUE	[0, 1]	3	0	0.1365	0.0835	17.9	
TanhWithDropout	73	0.9	1E-10	FALSE	0.8	0.2	0	0	500	Uniform	0.5	2	10000	FALSE	[-0.5, 05]	3	0	0.1139	0.0653	213.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.8	0.8	1	1	500	Normal	0.5	0.5	1	TRUE	[0, 1]	3	0	0.1866	0.1383	58.8	
RectifierWithDropout	219	0.999	1E-10	TRUE	0.2	0.2	1	0	5	Normal	0.05	2	10000	TRUE	[0, 1]	3	0	0.1225	0.0733	34.9	
RectifierWithDropout	73	0.999	1E-10	TRUE	0.8	0.8	0	0	5	Uniform	0.05	2	10000	FALSE	[0, 1]	3	0	0.1273	0.0702	13.7	
RectifierWithDropout	219	0.9	0.000001	FALSE	0.8	0.8	0	1	5	Uniform Adaptive	0.05	0.5	10000	TRUE	[-0.5, 05]	1	0	0.1999	0.2427	59.1	
TanhWithDropout	219	0.9	0.000001	TRUE	0.8	0.2	0	1	5	Uniform Adaptive	0.5	0.5	10000	TRUE	[0, 1]	2	0	0.1249	0.0766	130.2	
TanhWithDropout	219	0.9	0.000001	FALSE	0.2	0.8	0	0	500	Normal	0.05	2	1	FALSE	[-0.5, 05]	1	0	0.0996	0.0458	106.6	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.2	0.8	0	0	500	Uniform	0.05	2	1	FALSE	[-0.5, 05]	2	0	0.1990	0.2470	37.9	

TanhWithDropout	73	0.9	1E-10	FALSE	0.2	0.2	1	1	500	Normal	0.05	0.5	1	FALSE	[0, 1]	1	0	0.1863	0.1413	48.0	
RectifierWithDropout	219	0.9	0.000001	TRUE	0.2	0.2	1	0	5	Uniform Adaptive	0.5	0.5	10000	FALSE	[-0.5, 05]	1	0	0.1999	0.2509	157.0	
TanhWithDropout	73	0.999	0.000001	TRUE	0.8	0.8	0	1	5	Uniform Adaptive	0.05	2	10000	TRUE	[-0.5, 05]	2	0	0.1211	0.0858	67.7	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.8	0.8	1	1	5	Normal	0.5	0.5	10000	TRUE	[0, 1]	2	0				Exp Growth
TanhWithDropout	219	0.9	1E-10	TRUE	0.2	0.8	1	0	500	Uniform	0.05	0.5	1	TRUE	[-0.5, 05]	3	0	0.1286	0.1284	107.2	
RectifierWithDropout	73	0.999	0.000001	FALSE	0.8	0.2	0	0	5	Uniform Adaptive	0.05	0.5	10000	TRUE	[0, 1]	3	0				Exp Growth
RectifierWithDropout	219	0.999	0.000001	TRUE	0.8	0.2	1	1	500	Uniform	0.05	2	10000	FALSE	[-0.5, 05]	2	0				Exp Growth
RectifierWithDropout	73	0.9	0.000001	FALSE	0.8	0.2	0	0	500	Uniform	0.5	0.5	1	FALSE	[0, 1]	1	0	0.1850	0.1779	23.9	
RectifierWithDropout	73	0.999	0.000001	TRUE	0.2	0.8	1	1	5	Uniform Adaptive	0.05	2	10000	FALSE	[-0.5, 05]	1	0	0.1999	0.2427	26.3	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1630	0.1136	99.3	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1733	0.0776	63.4	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1607	0.1130	62.1	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.2061	0.0816	74.3	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1396	0.1402	107.2	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1996	0.2426	62.0	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1647	0.0807	71.1	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1627	0.1126	61.4	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1364	0.0787	31.3	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1622	0.1129	123.8	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1972	0.2452	61.9	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1918	0.0751	72.4	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1644	0.1139	92.0	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1574	0.0817	47.2	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1329	0.0728	71.1	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1999	0.2427	57.8	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1277	0.0741	37.8	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1424	0.1401	128.9	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1614	0.1132	80.3	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1603	0.1120	87.0	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1971	0.2453	60.7	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1630	0.1152	76.1	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1261	0.0740	39.3	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1500	0.0757	65.5	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1407	0.1418	177.5	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1390	0.1405	81.6	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1400	0.1407	83.9	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1406	0.1406	82.2	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1509	0.0761	55.8	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1619	0.1125	91.9	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1649	0.1143	67.7	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1270	0.0747	25.7	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1279	0.0765	32.7	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1973	0.2452	63.7	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1608	0.1104	61.6	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1628	0.1135	79.4	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1342	0.0756	37.9	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1221	0.0768	51.4	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1624	0.1130	74.7	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1651	0.1146	105.0	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1427	0.1414	89.2	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1999	0.2427	63.7	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1697	0.0793	73.1	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1990	0.2469	53.5	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1656	0.1149	68.1	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1424	0.1412	82.5	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1268	0.0763	26.2	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1392	0.1404	120.4	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1640	0.1133	67.6	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.2636	0.0814	70.5	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1989	0.2470	78.0	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1392	0.1394	98.7	
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1739	0.0738	61.5	
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1412	0.1454	81.5	
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1641	0.1137	104.0	
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1308	0.0813	52.3	

RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1998	0.2427	97.1
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1990	0.2470	63.4
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1365	0.0755	29.2
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1625	0.1132	108.8
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1426	0.1416	144.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1643	0.1149	80.5
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1999	0.2427	60.3
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1997	0.2427	115.9
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1390	0.1379	104.2
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1990	0.2470	61.9
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1973	0.2452	64.1
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1426	0.1404	80.5
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1990	0.2470	88.7
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1625	0.1148	67.5
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1401	0.1417	171.5
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1973	0.2452	66.4
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1990	0.2470	69.8
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1343	0.0774	26.8
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1291	0.0752	26.2
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1428	0.1427	81.2
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1973	0.2452	70.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1401	0.1397	91.1
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1626	0.1147	61.6
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1394	0.1426	79.5
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1407	0.1399	175.6
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1597	0.1106	69.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1637	0.1143	61.2
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1620	0.1126	85.8
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1624	0.1142	97.9
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1323	0.0782	38.3
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1391	0.1390	145.6
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1653	0.1148	85.4
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1398	0.1380	144.1
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1973	0.2453	69.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1619	0.1136	74.8
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1401	0.1410	89.2
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1392	0.1411	80.6
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1990	0.2470	63.3
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1362	0.0740	39.0
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1407	0.1405	81.0
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1994	0.0765	55.9
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1987	0.2470	62.5
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1392	0.1405	89.6
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	3	1	0.1971	0.2452	61.3
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1318	0.0783	31.6
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1626	0.1146	62.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1643	0.1137	129.9
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1423	0.1399	121.0
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1426	0.1408	104.3
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1371	0.0742	57.9
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1972	0.2452	103.7
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1997	0.2428	71.2
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1620	0.1143	92.4
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1998	0.2427	76.3
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1973	0.2452	62.8
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1401	0.1383	80.5
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1990	0.2470	64.4
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1431	0.1412	80.4
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1321	0.0777	25.7
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1989	0.2470	83.8
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1392	0.1402	80.8
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1392	0.0721	24.1
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1305	0.0775	46.2
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1998	0.2427	55.6
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1427	0.1391	82.0

RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1328	0.0774	26.8
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1310	0.0753	25.1
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1990	0.2470	61.1
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1648	0.1141	94.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	2	1	0.1624	0.1129	85.5
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1402	0.1378	83.0
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1420	0.1391	106.2
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1970	0.2452	60.7
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	3	1	0.1973	0.2452	73.3
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	2	1	0.1631	0.1138	68.5
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1420	0.1390	113.7
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1998	0.2427	60.0
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1389	0.1412	152.8
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1667	0.0739	72.9
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	1	1	0.1620	0.1139	111.0
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[0, 1]	3	1	0.1809	0.0761	52.3
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform Adaptive	0.275	1.25	5000	FALSE	[0, 1]	1	1	0.1632	0.1155	62.2
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1375	0.0708	33.6
RectifierWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	FALSE	[-0.5, 05]	1	1	0.1999	0.2439	56.4
TanhWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	FALSE	[-0.5, 05]	2	1	0.1392	0.1406	96.7
TanhWithDropout	146	0.99	1E-08	TRUE	0.5	0.5	0.5	0.5	252.5	Normal	0.275	1.25	5000	TRUE	[0, 1]	3	1	0.1650	0.1145	85.4
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	2	1	0.1985	0.2468	74.2
RectifierWithDropout	146	0.99	1E-08	FALSE	0.5	0.5	0.5	0.5	252.5	Uniform	0.275	1.25	5000	TRUE	[-0.5, 05]	1	1	0.1999	0.2427	61.2

Appendix O: CCD Phase I

Table 32. CCD Phase 1 Test Matrix

Neurons Per Hidden Layer	Rho	Epsilon	Input Dropout Rate	Hidden Dropout Rate	L1	L2	Max W2	Average Activation	Sparsity Beta	Minibatch Size	Test MSE RectifierWithDropout Test/Train 1	Test MSE RectifierWithDropout Test/Train 2	Test MSE RectifierWithDropout Test/Train 3	Test MSE TanhWithDropout Test/Train 1	Test MSE TanhWithDropout Test/Train 2	Test MSE TanhWithDropout Test/Train 3
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	1	0.1217	0.1216	0.1180	0.1156	0.1122	0.1179
52	0.8702691	2.23E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	2.405396	0.020812	0.20812	3513	0.1113	0.1099	0.1122	0.1111	0.1074	0.1125
32	0.9297289	4.1E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	2.405396	0.020812	0.20812	3513	0.1174	0.1127	0.1165	0.1136	0.1098	0.1141
32	0.9297289	2.23E-13	0.0767563	0.0767563	0.000017	2.97E-06	2.405396	0.020812	0.20812	3513	0.1202	0.1163	0.1266	0.1190	0.1116	0.1167
52	0.8702691	4.1E-13	0.1332437	0.0767563	0.000017	2.97E-06	2.405396	0.020812	0.20812	3513	0.1063	0.1076	0.1081	0.1020	0.0990	0.1000
52	0.9297289	4.1E-13	0.0767563	0.0767563	2.97E-06	0.000017	2.405396	0.020812	0.20812	3513	0.1091	0.0982	0.1026	0.1070	0.0996	0.1061
32	0.8702691	2.23E-13	0.1332437	0.0767563	2.97E-06	0.000017	2.405396	0.020812	0.20812	3513	0.1215	0.1206	0.1303	0.1145	0.1091	0.1128
32	0.8702691	4.1E-13	0.0767563	0.1332437	0.000017	0.000017	2.405396	0.020812	0.20812	3513	0.1191	0.1176	0.1186	0.1177	0.1094	0.1137
52	0.9297289	2.23E-13	0.1332437	0.1332437	0.000017	0.000017	2.405396	0.020812	0.20812	3513	0.1096	0.1054	0.1067	0.1127	0.1091	0.1114
52	0.9297289	2.23E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	3.594604	0.020812	0.20812	3513	0.1221	0.1159	0.1281	0.1180	0.1134	0.1170
32	0.8702691	4.1E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	3.594604	0.020812	0.20812	3513	0.1252	0.1216	0.1290	0.1183	0.1099	0.1156
32	0.8702691	2.23E-13	0.0767563	0.1332437	0.000017	2.97E-06	3.594604	0.020812	0.20812	3513	0.1295	0.1280	0.1219	0.1207	0.1174	0.1228
52	0.9297289	4.1E-13	0.1332437	0.1332437	0.000017	2.97E-06	3.594604	0.020812	0.20812	3513	0.1177	0.1188	0.1172	0.1183	0.1174	0.1170
52	0.8702691	4.1E-13	0.0767563	0.1332437	2.97E-06	0.000017	3.594604	0.020812	0.20812	3513	0.1227	0.1098	0.1249	0.1184	0.1136	0.1202
32	0.9297289	2.23E-13	0.1332437	0.1332437	2.97E-06	0.000017	3.594604	0.020812	0.20812	3513	0.1280	0.1251	0.1402	0.1430	0.1129	0.1220
32	0.9297289	4.1E-13	0.0767563	0.0767563	0.000017	0.000017	3.594604	0.020812	0.20812	3513	0.1247	0.1181	0.1261	0.1220	0.1161	0.1176
52	0.8702691	2.23E-13	0.1332437	0.0767563	0.000017	0.000017	3.594604	0.020812	0.20812	3513	0.1431	0.1275	0.1216	0.1192	0.1158	0.1207
32	0.8702691	4.1E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	2.405396	0.034188	0.20812	3513	0.1204	0.1136	0.1261	0.1115	0.1135	0.1125
52	0.9297289	2.23E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	2.405396	0.034188	0.20812	3513	0.1125	0.1039	0.1206	0.1207	0.1120	0.1169
52	0.9297289	4.1E-13	0.0767563	0.0767563	0.000017	2.97E-06	2.405396	0.034188	0.20812	3513	0.1103	0.1038	0.1071	0.1109	0.0980	0.1069
32	0.8702691	2.23E-13	0.1332437	0.0767563	0.000017	2.97E-06	2.405396	0.034188	0.20812	3513	0.1296	0.1141	0.1168	0.1117	0.1090	0.1114
32	0.9297289	2.23E-13	0.0767563	0.0767563	2.97E-06	0.000017	2.405396	0.034188	0.20812	3513	0.1201	0.1113	0.1189	0.1143	0.1046	0.1112
52	0.8702691	4.1E-13	0.1332437	0.0767563	2.97E-06	0.000017	2.405396	0.034188	0.20812	3513	0.1092	0.1001	0.1148	0.1021	0.0966	0.1009
52	0.8702691	2.23E-13	0.0767563	0.1332437	0.000017	0.000017	2.405396	0.034188	0.20812	3513	0.1154	0.1082	0.1124	0.1109	0.1095	0.1111
32	0.9297289	4.1E-13	0.1332437	0.1332437	0.000017	0.000017	2.405396	0.034188	0.20812	3513	0.1179	0.1143	0.1208	0.1150	0.1072	0.1130
32	0.9297289	4.1E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	3.594604	0.034188	0.20812	3513	0.1282	0.1207	0.1233	0.1191	0.1157	0.1203
52	0.8702691	2.23E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	3.594604	0.034188	0.20812	3513	0.1228	0.1231	0.1302	0.1189	0.1147	0.1170
52	0.8702691	4.1E-13	0.0767563	0.1332437	0.000017	2.97E-06	3.594604	0.034188	0.20812	3513	0.1229	0.1187	0.1240	0.1156	0.1162	0.1189
32	0.9297289	2.23E-13	0.1332437	0.1332437	0.000017	2.97E-06	3.594604	0.034188	0.20812	3513	0.1278	0.1262	0.1308	0.1242	0.1144	0.1240
32	0.8702691	2.23E-13	0.0767563	0.1332437	2.97E-06	0.000017	3.594604	0.034188	0.20812	3513	0.1470	0.1252	0.1270	0.1207	0.1160	0.1207
52	0.9297289	4.1E-13	0.1332437	0.1332437	2.97E-06	0.000017	3.594604	0.034188	0.20812	3513	0.1246	0.1232	0.1336	0.1213	0.1182	0.1199
52	0.9297289	2.23E-13	0.0767563	0.0767563	0.000017	0.000017	3.594604	0.034188	0.20812	3513	0.1291	0.1164	0.1238	0.1185	0.1151	0.1107
32	0.8702691	4.1E-13	0.1332437	0.0767563	0.000017	0.000017	3.594604	0.034188	0.20812	3513	0.1287	0.1174	0.1252	0.1166	0.1133	0.1241
52	0.9297289	4.1E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	2.405396	0.020812	0.34188	3513	0.1053	0.1044	0.1083	0.1158	0.1011	0.1158
32	0.8702691	2.23E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	2.405396	0.020812	0.34188	3513	0.1234	0.1182	0.1189	0.1134	0.1100	0.1130
32	0.8702691	4.1E-13	0.0767563	0.0767563	0.000017	2.97E-06	2.405396	0.020812	0.34188	3513	0.1221	0.1114	0.1195	0.1089	0.1003	0.0979
52	0.9297289	2.23E-13	0.1332437	0.0767563	0.000017	2.97E-06	2.405396	0.020812	0.34188	3513	0.1123	0.1090	0.1089	0.1125	0.1042	0.1102
52	0.8702691	2.23E-13	0.0767563	0.0767563	2.97E-06	0.000017	2.405396	0.020812	0.34188	3513	0.1069	0.1059	0.1152	0.1148	0.1127	0.1045
32	0.9297289	4.1E-13	0.1332437	0.0767563	2.97E-06	0.000017	2.405396	0.020812	0.34188	3513	0.1301	0.1130	0.1184	0.1107	0.1019	0.1100

32	0.9297289	2.23E-13	0.0767563	0.1332437	0.000017	0.000017	2.405396	0.020812	0.34188	3513	0.1268	0.1122	0.1230	0.1219	0.1118	0.1124
52	0.8702691	4.1E-13	0.1332437	0.1332437	0.000017	0.000017	2.405396	0.020812	0.34188	3513	0.1111	0.1119	0.1054	0.1115	0.1055	0.1049
52	0.8702691	4.1E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	3.594604	0.020812	0.34188	3513	0.1199	0.1193	0.1192	0.1231	0.1096	0.1161
32	0.9297289	2.23E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	3.594604	0.020812	0.34188	3513	0.1285	0.1214	0.1245	0.1203	0.1153	0.1191
32	0.9297289	4.1E-13	0.0767563	0.1332437	0.000017	2.97E-06	3.594604	0.020812	0.34188	3513	0.1250	0.1202	0.1261	0.1235	0.1143	0.1222
52	0.8702691	2.23E-13	0.1332437	0.1332437	0.000017	2.97E-06	3.594604	0.020812	0.34188	3513	0.1225	0.1294	0.1334	0.1222	0.1170	0.1188
52	0.9297289	2.23E-13	0.0767563	0.1332437	2.97E-06	0.000017	3.594604	0.020812	0.34188	3513	0.1266	0.1206	0.1236	0.1183	0.1193	0.1196
32	0.8702691	4.1E-13	0.1332437	0.1332437	2.97E-06	0.000017	3.594604	0.020812	0.34188	3513	0.1230	0.1264	0.1248	0.1231	0.1142	0.1191
32	0.8702691	2.23E-13	0.0767563	0.0767563	0.000017	0.000017	3.594604	0.020812	0.34188	3513	0.1309	0.1340	0.1325	0.1289	0.1203	0.1257
52	0.9297289	4.1E-13	0.1332437	0.0767563	0.000017	0.000017	3.594604	0.020812	0.34188	3513	0.1205	0.1169	0.1181	0.1175	0.1120	0.1156
32	0.9297289	2.23E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	2.405396	0.034188	0.34188	3513	0.1201	0.1131	0.1161	0.1160	0.1084	0.1181
52	0.8702691	4.1E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	2.405396	0.034188	0.34188	3513	0.1106	0.1045	0.1126	0.1093	0.1033	0.1063
52	0.8702691	2.23E-13	0.0767563	0.0767563	0.000017	2.97E-06	2.405396	0.034188	0.34188	3513	0.1181	0.1092	0.1145	0.1147	0.1057	0.1113
32	0.9297289	4.1E-13	0.1332437	0.0767563	0.000017	2.97E-06	2.405396	0.034188	0.34188	3513	0.1166	0.1116	0.1159	0.1083	0.1025	0.1062
32	0.8702691	4.1E-13	0.0767563	0.0767563	2.97E-06	0.000017	2.405396	0.034188	0.34188	3513	0.1239	0.1192	0.1185	0.1139	0.0999	0.1046
52	0.9297289	2.23E-13	0.1332437	0.0767563	2.97E-06	0.000017	2.405396	0.034188	0.34188	3513	0.1138	0.1102	0.1112	0.1073	0.1065	0.1063
52	0.9297289	4.1E-13	0.0767563	0.1332437	0.000017	0.000017	2.405396	0.034188	0.34188	3513	0.1112	0.1005	0.1071	0.1187	0.0996	0.1058
32	0.8702691	2.23E-13	0.1332437	0.1332437	0.000017	0.000017	2.405396	0.034188	0.34188	3513	0.1219	0.1181	0.1183	0.1140	0.1092	0.1162
32	0.8702691	2.23E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	3.594604	0.034188	0.34188	3513	0.1331	0.1234	0.1262	0.1191	0.1160	0.1239
52	0.9297289	4.1E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	3.594604	0.034188	0.34188	3513	0.1260	0.1122	0.1331	0.1151	0.1068	0.1154
52	0.9297289	2.23E-13	0.0767563	0.1332437	0.000017	2.97E-06	3.594604	0.034188	0.34188	3513	0.1361	0.1136	0.1169	0.1192	0.1178	0.1221
32	0.8702691	4.1E-13	0.1332437	0.1332437	0.000017	2.97E-06	3.594604	0.034188	0.34188	3513	0.1282	0.1212	0.1273	0.1198	0.1148	0.1182
32	0.9297289	4.1E-13	0.0767563	0.1332437	2.97E-06	0.000017	3.594604	0.034188	0.34188	3513	0.1466	0.1214	0.1290	0.1197	0.1168	0.1202
52	0.8702691	2.23E-13	0.1332437	0.1332437	2.97E-06	0.000017	3.594604	0.034188	0.34188	3513	0.1269	0.1198	0.1200	0.1234	0.1179	0.1213
52	0.8702691	4.1E-13	0.0767563	0.0767563	0.000017	0.000017	3.594604	0.034188	0.34188	3513	0.1287	0.1143	0.1264	0.1206	0.1062	0.1193
32	0.9297289	2.23E-13	0.1332437	0.0767563	0.000017	0.000017	3.594604	0.034188	0.34188	3513	0.1266	0.1205	0.1314	0.1209	0.1140	0.1169
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.05	5000	0.1171	0.1190	0.1159	0.1178	0.1153	0.1120
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.005	0.275	5000	0.1217	0.1181	0.1213	0.1204	0.1186	0.1172
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	1	0.0275	0.275	5000	0.0895	0.0832	0.0860	0.0925	0.0865	0.0888
42	0.899999	3.16E-13	0.105	0.105	0.00001	0	3	0.0275	0.275	5000	0.1263	0.1184	0.1204	0.1213	0.1089	0.1184
42	0.899999	3.16E-13	0.105	0.105	0	0.00001	3	0.0275	0.275	5000	0.1247	0.1198	0.1211	0.1170	0.1133	0.1185
42	0.899999	3.16E-13	0.105	0.01	0.00001	0.00001	3	0.0275	0.275	5000	0.1277	0.1141	0.1174	0.1380	0.1298	0.1355
42	0.899999	3.16E-13	0.01	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1245	0.1152	0.1198	0.1166	0.1113	0.1198
42	0.899999	1E-15	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1657	0.1598	0.1609	0.2972	0.2853	0.3013
42	0.8	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1258	0.1202	0.1267	0.1150	0.1114	0.1117
10	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1444	0.1406	0.1462	0.1269	0.1181	0.1204
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1190	0.1119	0.1220	0.1194	0.1093	0.1177
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1241	0.1281	0.1244	0.1192	0.1100	0.1121
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1234	0.1193	0.1228	0.1207	0.1099	0.1132
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1292	0.1158	0.1214	0.1193	0.1102	0.1201
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1202	0.1158	0.1196	0.1185	0.1035	0.1131
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1233	0.1174	0.1209	0.1215	0.1092	0.1203
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1224	0.1184	0.1224	0.1176	0.1144	0.1186
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1264	0.1156	0.1209	0.1163	0.1081	0.1181
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1278	0.1169	0.1224	0.1183	0.1088	0.1126
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1163	0.1164	0.1205	0.1173	0.1097	0.1135
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1266	0.1243	0.1184	0.1153	0.1120	0.1130
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1240	0.1133	0.1174	0.1167	0.1142	0.1142
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1197	0.1186	0.1175	0.1232	0.1064	0.1175
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1183	0.1209	0.1167	0.1176	0.1072	0.1105
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1181	0.1170	0.1228	0.1200	0.1171	0.1165
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1284	0.1198	0.1233	0.1187	0.1108	0.1168
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1246	0.1167	0.1205	0.1146	0.1101	0.1159
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1258	0.1130	0.1182	0.1156	0.1115	0.1154
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1240	0.1162	0.1231	0.1176	0.1083	0.1162
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1182	0.1162	0.1200	0.1157	0.1129	0.1143
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1311	0.1193	0.1241	0.1162	0.1092	0.1103
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1226	0.1117	0.1190	0.1200	0.1039	0.1157

42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1268	0.1184	0.1181	0.1180	0.1134	0.1167
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1217	0.1153	0.1222	0.1144	0.1135	0.1131
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1196	0.1178	0.1264	0.1184	0.1097	0.1153
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1203	0.1165	0.1419	0.1167	0.1124	0.1154
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1228	0.1152	0.1221	0.1174	0.1073	0.1119
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1288	0.1137	0.1199	0.1205	0.1102	0.1158
73	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1078	0.0972	0.1064	0.1138	0.1084	0.1107
42	0.999999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1090	0.0942	0.1016	0.0879	0.0821	0.0866
42	0.899999	1E-10	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.0932	0.0951	0.1031	0.0837	0.0781	0.0822
42	0.899999	3.16E-13	0.2	0.105	0.00001	0.00001	3	0.0275	0.275	5000	0.1148	0.1223	0.1246	0.1173	0.1124	0.1108
42	0.899999	3.16E-13	0.105	0.2	0.00001	0.00001	3	0.0275	0.275	5000	0.1233	0.1170	0.1216	0.1211	0.1192	0.1229
42	0.899999	3.16E-13	0.105	0.105	0.0000001	0.00001	3	0.0275	0.275	5000	0.1209	0.1167	0.1412	0.1172	0.1117	0.1149
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.0000001	3	0.0275	0.275	5000	0.1166	0.1123	0.1163	0.1177	0.1081	0.1107
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	5	0.0275	0.275	5000	0.1356	0.1272	0.1488	0.1243	0.1208	0.1240
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.05	0.275	5000	0.1221	0.1170	0.1217	0.1176	0.1113	0.1141
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.5	5000	0.1202	0.1170	0.1177	0.1187	0.1118	0.1140
32	0.8702691	4.1E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	2.405396	0.020812	0.20812	6487	0.1235	0.1155	0.1201	0.1172	0.0999	0.1045
52	0.9297289	2.23E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	2.405396	0.020812	0.20812	6487	0.1176	0.1039	0.1110	0.1135	0.0999	0.1139
52	0.9297289	4.1E-13	0.0767563	0.1332437	0.000017	2.97E-06	2.405396	0.020812	0.20812	6487	0.1077	0.0985	0.1045	0.1162	0.1040	0.1124
32	0.8702691	2.23E-13	0.1332437	0.1332437	0.000017	2.97E-06	2.405396	0.020812	0.20812	6487	0.1255	0.1146	0.1257	0.1177	0.1071	0.1099
32	0.9297289	2.23E-13	0.0767563	0.1332437	2.97E-06	0.000017	2.405396	0.020812	0.20812	6487	0.1220	0.1195	0.1280	0.1187	0.1120	0.1157
52	0.8702691	4.1E-13	0.1332437	0.1332437	2.97E-06	0.000017	2.405396	0.020812	0.20812	6487	0.1111	0.1038	0.1210	0.1095	0.1026	0.1086
52	0.8702691	2.23E-13	0.0767563	0.0767563	0.000017	0.000017	2.405396	0.020812	0.20812	6487	0.1204	0.1097	0.1103	0.1078	0.1019	0.1053
32	0.9297289	4.1E-13	0.1332437	0.0767563	0.000017	0.000017	2.405396	0.020812	0.20812	6487	0.1155	0.1127	0.1204	0.1095	0.1043	0.1104
32	0.9297289	4.1E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	3.594604	0.020812	0.20812	6487	0.1278	0.1281	0.1284	0.1201	0.1188	0.1171
52	0.8702691	2.23E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	3.594604	0.020812	0.20812	6487	0.1389	0.1241	0.1272	0.1243	0.1187	0.1207
52	0.8702691	4.1E-13	0.0767563	0.0767563	0.000017	2.97E-06	3.594604	0.020812	0.20812	6487	0.1246	0.1172	0.1243	0.1180	0.1114	0.1055
32	0.9297289	2.23E-13	0.1332437	0.0767563	0.000017	2.97E-06	3.594604	0.020812	0.20812	6487	0.1265	0.1202	0.1329	0.1184	0.1181	0.1196
32	0.8702691	2.23E-13	0.0767563	0.0767563	2.97E-06	0.000017	3.594604	0.020812	0.20812	6487	0.1273	0.1278	0.1352	0.1285	0.1200	0.1246
52	0.9297289	4.1E-13	0.1332437	0.0767563	2.97E-06	0.000017	3.594604	0.020812	0.20812	6487	0.1138	0.1156	0.1203	0.1138	0.1100	0.1188
52	0.9297289	2.23E-13	0.0767563	0.1332437	0.000017	0.000017	3.594604	0.020812	0.20812	6487	0.1199	0.1183	0.1230	0.1187	0.1163	0.1233
32	0.8702691	4.1E-13	0.1332437	0.1332437	0.000017	0.000017	3.594604	0.020812	0.20812	6487	0.1248	0.1223	0.1270	0.1183	0.1104	0.1197
52	0.8702691	2.23E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	2.405396	0.034188	0.20812	6487	0.1137	0.1066	0.1092	0.1123	0.1014	0.1034
32	0.9297289	4.1E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	2.405396	0.034188	0.20812	6487	0.1167	0.1113	0.1168	0.1135	0.1042	0.1065
32	0.9297289	2.23E-13	0.0767563	0.1332437	0.000017	2.97E-06	2.405396	0.034188	0.20812	6487	0.1250	0.1178	0.1243	0.1146	0.1126	0.1136
52	0.8702691	4.1E-13	0.1332437	0.1332437	0.000017	2.97E-06	2.405396	0.034188	0.20812	6487	0.1202	0.1045	0.1084	0.1120	0.1083	0.1134
52	0.9297289	4.1E-13	0.0767563	0.1332437	2.97E-06	0.000017	2.405396	0.034188	0.20812	6487	0.1069	0.1000	0.1114	0.1140	0.1066	0.1106
32	0.8702691	2.23E-13	0.1332437	0.1332437	2.97E-06	0.000017	2.405396	0.034188	0.20812	6487	0.1285	0.1165	0.1218	0.1195	0.1141	0.1205
32	0.8702691	4.1E-13	0.0767563	0.0767563	0.000017	0.000017	2.405396	0.034188	0.20812	6487	0.1260	0.1153	0.1186	0.1076	0.1002	0.1057
52	0.9297289	2.23E-13	0.1332437	0.0767563	0.000017	0.000017	2.405396	0.034188	0.20812	6487	0.1108	0.1021	0.1068	0.1110	0.0995	0.1070
52	0.9297289	2.23E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	3.594604	0.034188	0.20812	6487	0.1256	0.1236	0.1260	0.1224	0.1190	0.1200
32	0.8702691	4.1E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	3.594604	0.034188	0.20812	6487	0.1328	0.1266	0.1239	0.1200	0.1156	0.1199
32	0.8702691	2.23E-13	0.0767563	0.0767563	0.000017	2.97E-06	3.594604	0.034188	0.20812	6487	0.1516	0.1276	0.1276	0.1220	0.1177	0.1223
52	0.9297289	4.1E-13	0.1332437	0.0767563	0.000017	2.97E-06	3.594604	0.034188	0.20812	6487	0.1237	0.1115	0.1217	0.1185	0.1122	0.1187
52	0.8702691	4.1E-13	0.0767563	0.0767563	2.97E-06	0.000017	3.594604	0.034188	0.20812	6487	0.1384	0.1167	0.1206	0.1171	0.1078	0.1172
32	0.9297289	2.23E-13	0.1332437	0.0767563	2.97E-06	0.000017	3.594604	0.034188	0.20812	6487	0.1261	0.1222	0.1250	0.1206	0.1173	0.1192
32	0.9297289	4.1E-13	0.0767563	0.1332437	0.000017	0.000017	3.594604	0.034188	0.20812	6487	0.1243	0.1220	0.1210	0.1427	0.1168	0.1198
52	0.8702691	2.23E-13	0.1332437	0.1332437	0.000017	0.000017	3.594604	0.034188	0.20812	6487	0.1210	0.1170	0.1192	0.1227	0.1199	0.1228
32	0.9297289	2.23E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	2.405396	0.020812	0.34188	6487	0.1223	0.1190	0.1177	0.1113	0.1043	0.1116
52	0.8702691	4.1E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	2.405396	0.020812	0.34188	6487	0.1053	0.1042	0.1079	0.1044	0.0948	0.1099
52	0.8702691	2.23E-13	0.0767563	0.1332437	0.000017	2.97E-06	2.405396	0.020812	0.34188	6487	0.1177	0.1056	0.1130	0.1174	0.1092	0.1079
32	0.9297289	4.1E-13	0.1332437	0.1332437	0.000017	2.97E-06	2.405396	0.020812	0.34188	6487	0.1170	0.1102	0.1159	0.1191	0.1070	0.1119
32	0.8702691	4.1E-13	0.0767563	0.1332437	2.97E-06	0.000017	2.405396	0.020812	0.34188	6487	0.1251	0.1152	0.1160	0.1173	0.1033	0.1129
52	0.9297289	2.23E-13	0.1332437	0.1332437	2.97E-06	0.000017	2.405396	0.020812	0.34188	6487	0.1110	0.1064	0.1132	0.1167	0.1028	0.1180
52	0.9297289	4.1E-13	0.0767563	0.0767563	0.000017	0.000017	2.405396	0.020812	0.34188	6487	0.1055	0.1003	0.1002	0.1140	0.0996	0.1048
32	0.8702691	2.23E-13	0.1332437	0.0767563	0.000017	0.000017	2.405396	0.020812	0.34188	6487	0.1189	0.1152	0.1231	0.1120	0.1078	0.1120
32	0.8702691	2.23E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	3.594604	0.020812	0.34188	6487	0.1282	0.1248	0.1271	0.1217	0.1170	0.1238
52	0.9297289	4.1E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	3.594604	0.020812	0.34188	6487	0.1186	0.1125	0.1155	0.1181	0.1115	0.1183

52	0.9297289	2.23E-13	0.0767563	0.0767563	0.000017	2.97E-06	3.594604	0.020812	0.34188	6487	0.1269	0.1137	0.1251	0.1233	0.1161	0.1150
32	0.8702691	4.1E-13	0.1332437	0.0767563	0.000017	2.97E-06	3.594604	0.020812	0.34188	6487	0.1266	0.1216	0.1335	0.1211	0.1106	0.1169
32	0.9297289	4.1E-13	0.0767563	0.0767563	2.97E-06	0.000017	3.594604	0.020812	0.34188	6487	0.1244	0.1257	0.1249	0.1195	0.1128	0.1168
52	0.8702691	2.23E-13	0.1332437	0.0767563	2.97E-06	0.000017	3.594604	0.020812	0.34188	6487	0.1211	0.1211	0.1204	0.1291	0.1075	0.1217
52	0.8702691	4.1E-13	0.0767563	0.1332437	0.000017	0.000017	3.594604	0.020812	0.34188	6487	0.1269	0.1137	0.1201	0.1176	0.1129	0.1179
32	0.9297289	2.23E-13	0.1332437	0.1332437	0.000017	0.000017	3.594604	0.020812	0.34188	6487	0.1201	0.1218	0.1244	0.1222	0.1166	0.1236
52	0.9297289	4.1E-13	0.0767563	0.0767563	2.97E-06	2.97E-06	2.405396	0.034188	0.34188	6487	0.1030	0.1014	0.1027	0.1111	0.0970	0.1048
32	0.8702691	2.23E-13	0.1332437	0.0767563	2.97E-06	2.97E-06	2.405396	0.034188	0.34188	6487	0.1233	0.1234	0.1253	0.1210	0.1089	0.1121
32	0.8702691	4.1E-13	0.0767563	0.1332437	0.000017	2.97E-06	2.405396	0.034188	0.34188	6487	0.1209	0.1149	0.1205	0.1166	0.1100	0.1102
52	0.9297289	2.23E-13	0.1332437	0.1332437	0.000017	2.97E-06	2.405396	0.034188	0.34188	6487	0.1171	0.1078	0.1102	0.1176	0.1063	0.1126
52	0.8702691	2.23E-13	0.0767563	0.1332437	2.97E-06	0.000017	2.405396	0.034188	0.34188	6487	0.1142	0.1056	0.1148	0.1153	0.1053	0.1100
32	0.9297289	4.1E-13	0.1332437	0.1332437	2.97E-06	0.000017	2.405396	0.034188	0.34188	6487	0.1160	0.1131	0.1150	0.1170	0.1086	0.1129
32	0.9297289	2.23E-13	0.0767563	0.0767563	0.000017	0.000017	2.405396	0.034188	0.34188	6487	0.1331	0.1140	0.1281	0.1210	0.1050	0.1140
52	0.8702691	4.1E-13	0.1332437	0.0767563	0.000017	0.000017	2.405396	0.034188	0.34188	6487	0.1074	0.1018	0.1103	0.1004	0.0925	0.1013
52	0.8702691	4.1E-13	0.0767563	0.1332437	2.97E-06	2.97E-06	3.594604	0.034188	0.34188	6487	0.1269	0.1182	0.1246	0.1211	0.1172	0.1217
32	0.9297289	2.23E-13	0.1332437	0.1332437	2.97E-06	2.97E-06	3.594604	0.034188	0.34188	6487	0.1264	0.1282	0.1351	0.1218	0.1187	0.1247
32	0.9297289	4.1E-13	0.0767563	0.0767563	0.000017	2.97E-06	3.594604	0.034188	0.34188	6487	0.1247	0.1231	0.1220	0.1176	0.1174	0.1199
52	0.8702691	2.23E-13	0.1332437	0.0767563	0.000017	2.97E-06	3.594604	0.034188	0.34188	6487	0.1316	0.1162	0.1211	0.1268	0.1119	0.1098
52	0.9297289	2.23E-13	0.0767563	0.0767563	2.97E-06	0.000017	3.594604	0.034188	0.34188	6487	0.1243	0.1215	0.1221	0.1242	0.1175	0.1206
32	0.8702691	4.1E-13	0.1332437	0.0767563	2.97E-06	0.000017	3.594604	0.034188	0.34188	6487	0.1258	0.1219	0.1380	0.1202	0.1162	0.1183
32	0.8702691	2.23E-13	0.0767563	0.1332437	0.000017	0.000017	3.594604	0.034188	0.34188	6487	0.1278	0.1250	0.1253	0.1212	0.1152	0.1245
52	0.9297289	4.1E-13	0.1332437	0.1332437	0.000017	0.000017	3.594604	0.034188	0.34188	6487	0.1216	0.1131	0.1174	0.1210	0.1157	0.1189
42	0.899999	3.16E-13	0.105	0.105	0.00001	0.00001	3	0.0275	0.275	10000	0.1208	0.1204	0.1194	0.1201	0.1141	0.1198

Appendix P: CCD Phase II

Table 33.CCD Phase II Test Matrix

Neurons Per Hidden Layer	Rho	Epsilon	Input Dropout Rate	Hidden Dropout Rate	L1	L2	Max W2	Average Activation	Sparsity Beta	Minibatch Size	Test MSE RectifierWithDropout Test/Train 1	Test MSE RectifierWithDropout Test/Train 2	Test MSE RectifierWithDropout Test/Train 3	Test MSE TanhWithDropout Test/Train 1	Test MSE TanhWithDropout Test/Train 2	Test MSE TanhWithDropout Test/Train 3
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	1	0.0879	0.0753	0.0805	0.0754	0.0693	0.0728
93	0.76	8.47E-11	0.0081	0.01	0.000001	0.000001	0.6	0.020812	0.20812	3513	0.0847	0.0748	0.0792	0.0761	0.0720	0.0754
73	0.8	1.71E-10	0.01	0.01	0.000001	0.000001	0.6	0.020812	0.20812	3513	0.0898	0.0748	0.0828	0.0777	0.0742	0.0776
73	0.8	8.47E-11	0.0081	0.0081	0.0001	0.000001	0.6	0.020812	0.20812	3513	0.0850	0.0759	0.0826	0.0786	0.0742	0.0776
93	0.76	1.71E-10	0.01	0.0081	0.0001	0.000001	0.6	0.020812	0.20812	3513	0.0899	0.0732	0.0810	0.0761	0.0721	0.0755
93	0.8	1.71E-10	0.0081	0.0081	0.000001	0.0001	0.6	0.020812	0.20812	3513	0.0906	0.0726	0.0815	0.0757	0.0713	0.0751
73	0.76	8.47E-11	0.01	0.0081	0.000001	0.0001	0.6	0.020812	0.20812	3513	0.0861	0.0762	0.0842	0.0787	0.0746	0.0779
73	0.76	1.71E-10	0.0081	0.01	0.0001	0.0001	0.6	0.020812	0.20812	3513	0.0869	0.0752	0.0821	0.0786	0.0747	0.0781
93	0.8	8.47E-11	0.01	0.01	0.0001	0.0001	0.6	0.020812	0.20812	3513	0.0840	0.0740	0.0789	0.0756	0.0716	0.0752
93	0.8	8.47E-11	0.0081	0.0081	0.000001	0.000001	1	0.020812	0.20812	3513	0.0843	0.0707	0.0783	0.0700	0.0651	0.0693
73	0.76	1.71E-10	0.01	0.0081	0.000001	0.000001	1	0.020812	0.20812	3513	0.0903	0.0722	0.0820	0.0738	0.0688	0.0729
73	0.76	8.47E-11	0.0081	0.01	0.0001	0.000001	1	0.020812	0.20812	3513	0.0875	0.0728	0.0820	0.0741	0.0689	0.0734
93	0.8	1.71E-10	0.01	0.01	0.0001	0.000001	1	0.020812	0.20812	3513	0.0863	0.0699	0.0797	0.0704	0.0657	0.0692
93	0.76	1.71E-10	0.0081	0.01	0.000001	0.0001	1	0.020812	0.20812	3513	0.0842	0.0705	0.0804	0.0721	0.0668	0.0699
73	0.8	8.47E-11	0.01	0.01	0.000001	0.0001	1	0.020812	0.20812	3513	0.0834	0.0716	0.0824	0.0728	0.0692	0.0723
73	0.8	1.71E-10	0.0081	0.0081	0.0001	0.0001	1	0.020812	0.20812	3513	0.0849	0.0724	0.0816	0.0731	0.0682	0.0720
93	0.76	8.47E-11	0.01	0.0081	0.0001	0.0001	1	0.020812	0.20812	3513	0.0834	0.0707	0.0780	0.0720	0.0663	0.0712
73	0.76	1.71E-10	0.0081	0.01	0.000001	0.000001	0.6	0.034188	0.20812	3513	0.0873	0.0747	0.0821	0.0786	0.0747	0.0779
93	0.8	8.47E-11	0.01	0.01	0.000001	0.000001	0.6	0.034188	0.20812	3513	0.0826	0.0738	0.0803	0.0755	0.0715	0.0751
93	0.8	1.71E-10	0.0081	0.0081	0.0001	0.000001	0.6	0.034188	0.20812	3513	0.0901	0.0730	0.0814	0.0758	0.0715	0.0751
73	0.76	8.47E-11	0.01	0.0081	0.0001	0.000001	0.6	0.034188	0.20812	3513	0.0842	0.0757	0.0818	0.0789	0.0746	0.0782
73	0.8	8.47E-11	0.0081	0.0081	0.000001	0.0001	0.6	0.034188	0.20812	3513	0.0828	0.0763	0.0811	0.0783	0.0741	0.0777
93	0.76	1.71E-10	0.01	0.0081	0.000001	0.0001	0.6	0.034188	0.20812	3513	0.0901	0.0734	0.0806	0.0761	0.0720	0.0756
93	0.76	8.47E-11	0.0081	0.01	0.0001	0.0001	0.6	0.034188	0.20812	3513	0.0833	0.0743	0.0786	0.0762	0.0719	0.0758
73	0.8	1.71E-10	0.01	0.01	0.0001	0.0001	0.6	0.034188	0.20812	3513	0.0901	0.0742	0.0808	0.0783	0.0741	0.0778
73	0.8	1.71E-10	0.0081	0.0081	0.000001	0.000001	1	0.034188	0.20812	3513	0.0884	0.0725	0.0826	0.0745	0.0684	0.0723
93	0.76	8.47E-11	0.01	0.0081	0.000001	0.000001	1	0.034188	0.20812	3513	0.0829	0.0709	0.0790	0.0714	0.0669	0.0711
93	0.76	1.71E-10	0.0081	0.01	0.0001	0.000001	1	0.034188	0.20812	3513	0.0829	0.0698	0.0780	0.0716	0.0662	0.0707
73	0.8	8.47E-11	0.01	0.01	0.0001	0.000001	1	0.034188	0.20812	3513	0.0876	0.0719	0.0825	0.0733	0.0684	0.0720
73	0.76	8.47E-11	0.0081	0.01	0.000001	0.0001	1	0.034188	0.20812	3513	0.0950	0.0735	0.0812	0.0747	0.0689	0.0733
93	0.8	1.71E-10	0.01	0.01	0.000001	0.0001	1	0.034188	0.20812	3513	0.0849	0.0711	0.0805	0.0708	0.0659	0.0696
93	0.8	8.47E-11	0.0081	0.0081	0.0001	0.0001	1	0.034188	0.20812	3513	0.0836	0.0690	0.0811	0.0698	0.0657	0.0691
73	0.76	1.71E-10	0.01	0.0081	0.0001	0.0001	1	0.034188	0.20812	3513	0.0853	0.0729	0.0867	0.0738	0.0687	0.0729
93	0.8	1.71E-10	0.0081	0.01	0.000001	0.000001	0.6	0.020812	0.34188	3513	0.1049	0.0774	0.0891	0.0758	0.0717	0.0754
73	0.76	8.47E-11	0.01	0.01	0.000001	0.000001	0.6	0.020812	0.34188	3513	0.0925	0.0795	0.0871	0.0793	0.0747	0.0783
73	0.76	1.71E-10	0.0081	0.0081	0.0001	0.000001	0.6	0.020812	0.34188	3513	0.1026	0.0794	0.0893	0.0792	0.0746	0.0782
93	0.8	8.47E-11	0.01	0.0081	0.0001	0.000001	0.6	0.020812	0.34188	3513	0.0947	0.0771	0.0858	0.0763	0.0718	0.0753
93	0.76	8.47E-11	0.0081	0.0081	0.000001	0.0001	0.6	0.020812	0.34188	3513	0.0916	0.0778	0.0856	0.0767	0.0721	0.0758
73	0.8	1.71E-10	0.01	0.0081	0.000001	0.0001	0.6	0.020812	0.34188	3513	0.0998	0.0787	0.0890	0.0786	0.0740	0.0778

73	0.8	8.47E-11	0.0081	0.01	0.0001	0.0001	0.6	0.020812	0.34188	3513	0.0897	0.0787	0.0866	0.0787	0.0744	0.0781
93	0.76	1.71E-10	0.01	0.01	0.0001	0.0001	0.6	0.020812	0.34188	3513	0.1023	0.0774	0.0870	0.0771	0.0723	0.0762
93	0.76	1.71E-10	0.0081	0.0081	0.000001	0.000001	1	0.020812	0.34188	3513	0.0977	0.0789	0.0817	0.0716	0.0666	0.0702
73	0.8	8.47E-11	0.01	0.0081	0.000001	0.000001	1	0.020812	0.34188	3513	0.0903	0.0759	0.0853	0.0751	0.0682	0.0723
73	0.8	1.71E-10	0.0081	0.01	0.0001	0.000001	1	0.020812	0.34188	3513	0.0928	0.0781	0.0877	0.0747	0.0685	0.0720
93	0.76	8.47E-11	0.01	0.01	0.0001	0.000001	1	0.020812	0.34188	3513	0.0872	0.0743	0.0845	0.0722	0.0664	0.0707
93	0.8	8.47E-11	0.0081	0.01	0.000001	0.0001	1	0.020812	0.34188	3513	0.0880	0.0745	0.0869	0.0728	0.0656	0.0705
73	0.76	1.71E-10	0.01	0.01	0.000001	0.0001	1	0.020812	0.34188	3513	0.0900	0.0793	0.0855	0.0744	0.0698	0.0740
73	0.76	8.47E-11	0.0081	0.0081	0.0001	0.0001	1	0.020812	0.34188	3513	0.0933	0.0764	0.0919	0.0753	0.0698	0.0748
93	0.8	1.71E-10	0.01	0.0081	0.0001	0.0001	1	0.020812	0.34188	3513	0.0983	0.0762	0.0843	0.0707	0.0657	0.0697
73	0.8	8.47E-11	0.0081	0.01	0.000001	0.000001	0.6	0.034188	0.34188	3513	0.0899	0.0793	0.0867	0.0792	0.0743	0.0781
93	0.76	1.71E-10	0.01	0.01	0.000001	0.000001	0.6	0.034188	0.34188	3513	0.1025	0.0776	0.0879	0.0766	0.0721	0.0761
93	0.76	8.47E-11	0.0081	0.0081	0.0001	0.000001	0.6	0.034188	0.34188	3513	0.0939	0.0768	0.0852	0.0766	0.0721	0.0758
73	0.8	1.71E-10	0.01	0.0081	0.0001	0.000001	0.6	0.034188	0.34188	3513	0.1030	0.0779	0.0898	0.0788	0.0743	0.0779
73	0.76	1.71E-10	0.0081	0.0081	0.000001	0.0001	0.6	0.034188	0.34188	3513	0.1017	0.0793	0.0882	0.0793	0.0746	0.0782
93	0.8	8.47E-11	0.01	0.0081	0.000001	0.0001	0.6	0.034188	0.34188	3513	0.0909	0.0759	0.0843	0.0760	0.0716	0.0752
93	0.8	1.71E-10	0.0081	0.01	0.0001	0.0001	0.6	0.034188	0.34188	3513	0.1040	0.0773	0.0886	0.0769	0.0717	0.0752
73	0.76	8.47E-11	0.01	0.01	0.0001	0.0001	0.6	0.034188	0.34188	3513	0.0919	0.0802	0.0850	0.0791	0.0750	0.0784
73	0.76	8.47E-11	0.0081	0.0081	0.000001	0.000001	1	0.034188	0.34188	3513	0.0950	0.0754	0.0881	0.0749	0.0695	0.0749
93	0.8	1.71E-10	0.01	0.0081	0.000001	0.000001	1	0.034188	0.34188	3513	0.0918	0.0745	0.0811	0.0718	0.0656	0.0700
93	0.8	8.47E-11	0.0081	0.01	0.0001	0.000001	1	0.034188	0.34188	3513	0.0837	0.0740	0.0828	0.0722	0.0662	0.0697
73	0.76	1.71E-10	0.01	0.01	0.0001	0.000001	1	0.034188	0.34188	3513	0.0944	0.0763	0.0877	0.0747	0.0697	0.0738
73	0.8	1.71E-10	0.0081	0.01	0.000001	0.0001	1	0.034188	0.34188	3513	0.0950	0.0765	0.0880	0.0754	0.0686	0.0736
93	0.76	8.47E-11	0.01	0.01	0.000001	0.0001	1	0.034188	0.34188	3513	0.0868	0.0723	0.0873	0.0719	0.0673	0.0714
93	0.76	1.71E-10	0.0081	0.0081	0.0001	0.0001	1	0.034188	0.34188	3513	0.0962	0.0751	0.0843	0.0736	0.0660	0.0702
73	0.8	8.47E-11	0.01	0.0081	0.0001	0.0001	1	0.034188	0.34188	3513	0.0905	0.0765	0.0887	0.0747	0.0685	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.05	5000	0.0712	0.0617	0.0693	0.0733	0.0689	0.0726
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.005	0.275	5000	0.0884	0.0753	0.0813	0.0739	0.0692	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.1272829	0.0275	0.275	5000	0.1063	0.0951	0.1009	0.0990	0.0956	0.0991
83	0.78	1.2E-10	0.009	0.009	0.00001	4.33E-09	0.8	0.0275	0.275	5000	0.0902	0.0756	0.0825	0.0743	0.0691	0.0731
83	0.78	1.2E-10	0.009	0.009	4.33E-09	0.00001	0.8	0.0275	0.275	5000	0.0872	0.0761	0.0827	0.0735	0.0690	0.0730
83	0.78	1.2E-10	0.009	0.0024081	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0924	0.0747	0.0815	0.0742	0.0690	0.0729
83	0.78	1.2E-10	0.0024081	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0936	0.0763	0.0821	0.0755	0.0688	0.0736
83	0.78	7.23E-11	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0865	0.0737	0.0850	0.0737	0.0696	0.0732
83	0.7127283	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0886	0.0750	0.0835	0.0747	0.0701	0.0738
39	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0949	0.0808	0.0929	0.0841	0.0790	0.0826
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0911	0.0751	0.0810	0.0743	0.0693	0.0732
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0915	0.0754	0.0834	0.0742	0.0688	0.0728
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0875	0.0751	0.0794	0.0739	0.0696	0.0732
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0897	0.0757	0.0840	0.0743	0.0692	0.0728
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0918	0.0756	0.0800	0.0737	0.0695	0.0732
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0900	0.0758	0.0815	0.0749	0.0693	0.0729
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0892	0.0748	0.0804	0.0741	0.0693	0.0730
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0888	0.0757	0.0791	0.0744	0.0692	0.0729
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0881	0.0757	0.0816	0.0733	0.0690	0.0734
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0905	0.0753	0.0798	0.0749	0.0691	0.0729
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0913	0.0762	0.0849	0.0745	0.0689	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0865	0.0756	0.0820	0.0740	0.0693	0.0737
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0861	0.0770	0.0817	0.0742	0.0692	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0906	0.0748	0.0832	0.0745	0.0692	0.0730
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0865	0.0755	0.0816	0.0740	0.0691	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0871	0.0752	0.0830	0.0747	0.0688	0.0728
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0902	0.0756	0.0788	0.0749	0.0690	0.0729
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0889	0.0747	0.0802	0.0744	0.0693	0.0738
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0869	0.0758	0.0841	0.0750	0.0691	0.0735
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0840	0.0758	0.0846	0.0733	0.0690	0.0734
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0877	0.0750	0.0825	0.0746	0.0694	0.0733
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0882	0.0766	0.0817	0.0745	0.0694	0.0734

83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0889	0.0747	0.0805	0.0742	0.0692	0.0729
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0878	0.0754	0.0821	0.0745	0.0692	0.0735
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0865	0.0751	0.0806	0.0741	0.0693	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0878	0.0750	0.0810	0.0744	0.0692	0.0731
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0875	0.0750	0.0829	0.0750	0.0691	0.0733
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0903	0.0749	0.0809	0.0748	0.0691	0.0729
127	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0883	0.0702	0.0778	0.0707	0.0651	0.0690
83	0.8472717	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0900	0.0742	0.0811	0.0728	0.0680	0.0717
83	0.78	4.05E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.1054	0.0746	0.0859	0.0736	0.0690	0.0728
83	0.78	1.2E-10	0.0336359	0.009	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0846	0.0723	0.0809	0.0737	0.0694	0.0734
83	0.78	1.2E-10	0.009	0.0336359	0.00001	0.00001	0.8	0.0275	0.275	5000	0.0834	0.0747	0.0838	0.0749	0.0700	0.0737
83	0.78	1.2E-10	0.009	0.009	0.0231	0.00001	0.8	0.0275	0.275	5000	0.0819	0.0661	0.0761	0.0742	0.0711	0.0743
83	0.78	1.2E-10	0.009	0.009	0.00001	0.0231	0.8	0.0275	0.275	5000	0.0835	0.0709	0.0815	0.0713	0.0675	0.0709
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	1.4727171	0.0275	0.275	5000	0.0977	0.0729	0.0836	0.0792	0.0692	0.0735
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.05	0.275	5000	0.0858	0.0751	0.0804	0.0748	0.0690	0.0735
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.5	5000	0.1037	0.0780	0.0881	0.0773	0.0697	0.0741
73	0.76	1.71E-10	0.0081	0.0081	0.000001	0.000001	0.6	0.020812	0.20812	6487	0.0921	0.0768	0.0824	0.0784	0.0744	0.0779
93	0.8	8.47E-11	0.01	0.0081	0.000001	0.000001	0.6	0.020812	0.20812	6487	0.0832	0.0741	0.0790	0.0755	0.0715	0.0750
93	0.8	1.71E-10	0.0081	0.01	0.0001	0.000001	0.6	0.020812	0.20812	6487	0.0921	0.0726	0.0799	0.0758	0.0715	0.0751
73	0.76	8.47E-11	0.01	0.01	0.0001	0.000001	0.6	0.020812	0.20812	6487	0.0858	0.0757	0.0838	0.0786	0.0747	0.0782
73	0.8	8.47E-11	0.0081	0.01	0.000001	0.0001	0.6	0.020812	0.20812	6487	0.0849	0.0754	0.0815	0.0781	0.0741	0.0775
93	0.76	1.71E-10	0.01	0.01	0.000001	0.0001	0.6	0.020812	0.20812	6487	0.0909	0.0729	0.0810	0.0762	0.0719	0.0755
93	0.76	8.47E-11	0.0081	0.0081	0.0001	0.0001	0.6	0.020812	0.20812	6487	0.0839	0.0746	0.0793	0.0761	0.0720	0.0755
73	0.8	1.71E-10	0.01	0.0081	0.0001	0.0001	0.6	0.020812	0.20812	6487	0.0892	0.0747	0.0818	0.0785	0.0740	0.0776
73	0.8	1.71E-10	0.0081	0.01	0.000001	0.000001	1	0.020812	0.20812	6487	0.0853	0.0733	0.0807	0.0736	0.0681	0.0722
93	0.76	8.47E-11	0.01	0.01	0.000001	0.000001	1	0.020812	0.20812	6487	0.0844	0.0708	0.0812	0.0714	0.0661	0.0701
93	0.76	1.71E-10	0.0081	0.0081	0.0001	0.000001	1	0.020812	0.20812	6487	0.0821	0.0730	0.0803	0.0708	0.0659	0.0705
73	0.8	8.47E-11	0.01	0.0081	0.0001	0.000001	1	0.020812	0.20812	6487	0.0898	0.0714	0.0790	0.0735	0.0689	0.0725
73	0.76	8.47E-11	0.0081	0.0081	0.000001	0.0001	1	0.020812	0.20812	6487	0.0887	0.0721	0.0826	0.0745	0.0688	0.0729
93	0.8	1.71E-10	0.01	0.0081	0.000001	0.0001	1	0.020812	0.20812	6487	0.0821	0.0703	0.0794	0.0702	0.0655	0.0696
93	0.8	8.47E-11	0.0081	0.01	0.0001	0.0001	1	0.020812	0.20812	6487	0.0856	0.0722	0.0790	0.0710	0.0654	0.0692
73	0.76	1.71E-10	0.01	0.01	0.0001	0.0001	1	0.020812	0.20812	6487	0.0898	0.0722	0.0834	0.0729	0.0692	0.0734
93	0.76	8.47E-11	0.0081	0.0081	0.000001	0.000001	0.6	0.034188	0.20812	6487	0.0857	0.0747	0.0790	0.0758	0.0718	0.0756
73	0.8	1.71E-10	0.01	0.0081	0.000001	0.000001	0.6	0.034188	0.20812	6487	0.0877	0.0745	0.0813	0.0785	0.0741	0.0777
73	0.8	8.47E-11	0.0081	0.01	0.0001	0.000001	0.6	0.034188	0.20812	6487	0.0846	0.0755	0.0826	0.0783	0.0742	0.0778
93	0.76	1.71E-10	0.01	0.01	0.0001	0.000001	0.6	0.034188	0.20812	6487	0.0892	0.0724	0.0792	0.0762	0.0721	0.0757
93	0.8	1.71E-10	0.0081	0.01	0.000001	0.0001	0.6	0.034188	0.20812	6487	0.0913	0.0724	0.0798	0.0756	0.0715	0.0750
73	0.76	8.47E-11	0.01	0.01	0.000001	0.0001	0.6	0.034188	0.20812	6487	0.0860	0.0752	0.0829	0.0782	0.0747	0.0780
73	0.76	1.71E-10	0.0081	0.0081	0.0001	0.0001	0.6	0.034188	0.20812	6487	0.0880	0.0750	0.0821	0.0785	0.0746	0.0781
93	0.8	8.47E-11	0.01	0.0081	0.0001	0.0001	0.6	0.034188	0.20812	6487	0.0827	0.0745	0.0804	0.0759	0.0715	0.0751
93	0.8	8.47E-11	0.0081	0.01	0.000001	0.000001	1	0.034188	0.20812	6487	0.0845	0.0712	0.0805	0.0706	0.0652	0.0700
73	0.76	1.71E-10	0.01	0.01	0.000001	0.000001	1	0.034188	0.20812	6487	0.0849	0.0718	0.0838	0.0747	0.0696	0.0731
73	0.76	8.47E-11	0.0081	0.0081	0.0001	0.000001	1	0.034188	0.20812	6487	0.0884	0.0725	0.0862	0.0750	0.0695	0.0729
93	0.8	1.71E-10	0.01	0.0081	0.0001	0.000001	1	0.034188	0.20812	6487	0.0840	0.0698	0.0786	0.0707	0.0662	0.0695
93	0.76	1.71E-10	0.0081	0.0081	0.000001	0.0001	1	0.034188	0.20812	6487	0.0837	0.0709	0.0780	0.0715	0.0665	0.0705
73	0.8	8.47E-11	0.01	0.0081	0.000001	0.0001	1	0.034188	0.20812	6487	0.0883	0.0727	0.0816	0.0731	0.0676	0.0728
73	0.8	1.71E-10	0.0081	0.01	0.0001	0.0001	1	0.034188	0.20812	6487	0.0873	0.0716	0.0823	0.0727	0.0681	0.0719
93	0.76	8.47E-11	0.01	0.01	0.0001	0.0001	1	0.034188	0.20812	6487	0.0822	0.0712	0.0801	0.0713	0.0665	0.0704
73	0.8	8.47E-11	0.0081	0.0081	0.000001	0.000001	0.6	0.020812	0.34188	6487	0.0923	0.0792	0.0845	0.0787	0.0743	0.0779
93	0.76	1.71E-10	0.01	0.0081	0.000001	0.000001	0.6	0.020812	0.34188	6487	0.1007	0.0781	0.0887	0.0766	0.0722	0.0760
93	0.76	8.47E-11	0.0081	0.01	0.0001	0.000001	0.6	0.020812	0.34188	6487	0.0934	0.0777	0.0857	0.0766	0.0723	0.0761
73	0.8	1.71E-10	0.01	0.01	0.0001	0.000001	0.6	0.020812	0.34188	6487	0.1018	0.0791	0.0898	0.0779	0.0744	0.0779
73	0.76	1.71E-10	0.0081	0.01	0.000001	0.0001	0.6	0.020812	0.34188	6487	0.1030	0.0796	0.0907	0.0790	0.0746	0.0783
93	0.8	8.47E-11	0.01	0.01	0.000001	0.0001	0.6	0.020812	0.34188	6487	0.0908	0.0770	0.0831	0.0766	0.0718	0.0753
93	0.8	1.71E-10	0.0081	0.0081	0.0001	0.0001	0.6	0.020812	0.34188	6487	0.1040	0.0778	0.0892	0.0759	0.0717	0.0756
73	0.76	8.47E-11	0.01	0.0081	0.0001	0.0001	0.6	0.020812	0.34188	6487	0.0897	0.0790	0.0871	0.0788	0.0747	0.0783
73	0.76	8.47E-11	0.0081	0.01	0.000001	0.000001	1	0.020812	0.34188	6487	0.0929	0.0778	0.0879	0.0758	0.0696	0.0733
93	0.8	1.71E-10	0.01	0.01	0.000001	0.000001	1	0.020812	0.34188	6487	0.0932	0.0753	0.0824	0.0703	0.0656	0.0697

93	0.8	8.47E-11	0.0081	0.0081	0.0001	0.000001	1	0.020812	0.34188	6487	0.0876	0.0734	0.0869	0.0721	0.0654	0.0696
73	0.76	1.71E-10	0.01	0.0081	0.0001	0.000001	1	0.020812	0.34188	6487	0.0929	0.0776	0.0878	0.0742	0.0694	0.0733
73	0.8	1.71E-10	0.0081	0.0081	0.000001	0.0001	1	0.020812	0.34188	6487	0.0951	0.0766	0.0845	0.0762	0.0685	0.0735
93	0.76	8.47E-11	0.01	0.0081	0.000001	0.0001	1	0.020812	0.34188	6487	0.0925	0.0747	0.0861	0.0712	0.0669	0.0713
93	0.76	1.71E-10	0.0081	0.01	0.0001	0.0001	1	0.020812	0.34188	6487	0.0918	0.0757	0.0859	0.0721	0.0666	0.0718
73	0.8	8.47E-11	0.01	0.01	0.0001	0.0001	1	0.020812	0.34188	6487	0.0891	0.0755	0.0869	0.0742	0.0683	0.0727
93	0.8	1.71E-10	0.0081	0.0081	0.000001	0.000001	0.6	0.034188	0.34188	6487	0.1040	0.0774	0.0896	0.0769	0.0715	0.0751
73	0.76	8.47E-11	0.01	0.0081	0.000001	0.000001	0.6	0.034188	0.34188	6487	0.0922	0.0813	0.0852	0.0791	0.0748	0.0786
73	0.76	1.71E-10	0.0081	0.01	0.0001	0.000001	0.6	0.034188	0.34188	6487	0.1016	0.0785	0.0874	0.0797	0.0747	0.0786
93	0.8	8.47E-11	0.01	0.01	0.0001	0.000001	0.6	0.034188	0.34188	6487	0.0916	0.0769	0.0843	0.0766	0.0717	0.0754
93	0.76	8.47E-11	0.0081	0.01	0.000001	0.0001	0.6	0.034188	0.34188	6487	0.0920	0.0758	0.0834	0.0772	0.0721	0.0763
73	0.8	1.71E-10	0.01	0.01	0.000001	0.0001	0.6	0.034188	0.34188	6487	0.1007	0.0779	0.0886	0.0790	0.0743	0.0777
73	0.8	8.47E-11	0.0081	0.0081	0.0001	0.0001	0.6	0.034188	0.34188	6487	0.0917	0.0795	0.0842	0.0794	0.0743	0.0781
93	0.76	1.71E-10	0.01	0.0081	0.0001	0.0001	0.6	0.034188	0.34188	6487	0.1046	0.0771	0.0879	0.0769	0.0723	0.0760
93	0.76	1.71E-10	0.0081	0.01	0.000001	0.000001	1	0.034188	0.34188	6487	0.0914	0.0770	0.0833	0.0716	0.0672	0.0703
73	0.8	8.47E-11	0.01	0.01	0.000001	0.000001	1	0.034188	0.34188	6487	0.0950	0.0771	0.0858	0.0749	0.0689	0.0725
73	0.8	1.71E-10	0.0081	0.0081	0.0001	0.000001	1	0.034188	0.34188	6487	0.0946	0.0765	0.0900	0.0768	0.0683	0.0735
93	0.76	8.47E-11	0.01	0.0081	0.0001	0.000001	1	0.034188	0.34188	6487	0.0888	0.0751	0.0844	0.0705	0.0666	0.0713
93	0.8	8.47E-11	0.0081	0.0081	0.000001	0.0001	1	0.034188	0.34188	6487	0.0863	0.0755	0.0844	0.0743	0.0659	0.0695
73	0.76	1.71E-10	0.01	0.0081	0.000001	0.0001	1	0.034188	0.34188	6487	0.0964	0.0777	0.0890	0.0745	0.0696	0.0726
73	0.76	8.47E-11	0.0081	0.01	0.0001	0.0001	1	0.034188	0.34188	6487	0.0885	0.0754	0.0863	0.0752	0.0698	0.0742
93	0.8	1.71E-10	0.01	0.01	0.0001	0.0001	1	0.034188	0.34188	6487	0.0933	0.0730	0.0821	0.0710	0.0653	0.0695
83	0.78	1.2E-10	0.009	0.009	0.00001	0.00001	0.8	0.0275	0.275	10000	0.0881	0.0765	0.0805	0.0740	0.0695	0.0730

Bibliography

[1] A. Sood and R. Enbody, “U. S. Military Defense Systems : The Anatomy Of Cyber Espionage By Chinese Hackers,” *Georg. J. Int. Aff.*, 2016.

[2] ATT Security, “What Every CEO needs to know about cybersecurity Decoding the adversary,” *ATT Cybersecurity Insights*, vol. 1, pp. 1–36, 2015.

[3] US Department of Defense, “The DoD Cyber Strategy.” U.S. Department of Defense, p. 42, 2015.

[4] M. Alvarez, N. Bradley, P. Cobb, S. Craig, R. Iffert, L. Kessem, J. Kravitz, D. McMillen, and S. Moore, “IBM X-Force Threat Intelligence Index 2017 The Year of the Mega Breach,” no. March, pp. 1–30, 2017.

[5] Symantec, “Internet Security Threat Report,” 2017.

[6] B. Barrett, “Game of Thrones Leak Puts Unreleased Script and Other HBO Shows Online | WIRED,” *WIRED*, 2017. [Online]. Available: https://www.wired.com/story/game-of-thrones-leak-hbo-hack/. [Accessed: 01-Aug-2017].

[7] DoD, “FACT SHEET: THE DEPARTMENT OF DEFENSE (DOD) CYBER STRATEGY.” DoD, p. 2, 2015.

[8] H. E. Poston, “A brief taxonomy of intrusion detection strategies,” *Natl. Aerosp. Electron. Conf. Proc. IEEE*, pp. 255–263, 2013.

[9] R. J. Gutierrez, “A Tabulated Vector Approach For Log-Based Anomaly Detection,” Air Force Institute of Technology, 2017.

[10] M. Markou and S. Singh, “Novelty detection: A review - Part 1: Statistical approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[11] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: a review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[12] C. H. (John) Wu and J. D. Irwin, *Introduction to Computer Networks and Cybersecurity*. Boca Raton: CRC Press/Taylor & Francis Group, LLC, 2013.

[13] V. Kumar, “Parallel and Distributed Computing for Cybersecurity,” *IEEE Distrib. Syst. Online*, vol. 6, no. 10, pp. 1–9, 2005.

[14] S. Agrawal and J. Agrawal, “Survey on anomaly detection using data mining techniques,” *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 708–713, 2015.

[15] C. Manikopoulos and S. Papavassiliou, “Network intrusion and fault detection: A statistical anomaly approach,” *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 76–82, 2002.

[16] A. Ghosh, J. Wanken, and F. Charron, “Detecting anomalous and unknown intrusions against programs,” *Comput. Secur. Appl. Conf. 1998. Proceedings. 14th Annu.*, pp. 259–267, 1998.

[17] A. Patcha and J. M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[18] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion-detection systems,” *Comput. Networks*, vol. 31, no. 8, pp. 805–822, 1999.

[19] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology,” *Nist Spec. Publ.*, vol. 800–94, p. 127, 2007.

[20] K. Scarfone and P. Hoffman, “Guidelines on firewalls and firewall policy: recommendations of the National Institute of Standards and Technology,” *NIST Spec. Publ.*, p. 74, 2009.

[21] R. G. Bace, *Intrusion Detection*, 2nd ed. University of Michigan: Macmillan Technical Publiishing, 2000.

[22] D. E. Denning, “An Intrusion-Detection Model,” *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 222–232, 1987.

[23] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” *Proc. 2003 SIAM Int. Conf. Data Min.*, pp. 25–36, 2003.

[24] J. R. Yost, “The March of IDES: Early History of Intrusion-Detection Expert Systems,” *IEEE Ann. Hist. Comput.*, vol. 38, no. 4, pp. 42–54, 2016.

[25] A. K. Marnerides, A. Schaeffer-Filho, and A. Mauthe, “Traffic anomaly diagnosis in Internet backbone networks: A survey,” *Comput. Networks*, vol. 73, pp. 224–243, 2014.

[26] “theatre,” *Cambridge Dictionary*. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/theatre?a=british. [Accessed: 08-Nov-2017].

[27] D. Swift, “A Practical Application of SIM/SEM/SIEM Automating Threat Identification.” SANS Institute InfoSec Reading Room, p. 39, 2001.

[28] T. J. Bihl, W. A. Young II, and G. R. Weckman, “Defining, Understanding, and Addressing Big Data,” *Int. J. Bus. Anal.*, vol. 3, no. 2, pp. 1–32, 2016.

[29] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, 2009.

[30] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, “Intrusion detection by machine learning: A review,” *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[31] H. Debar, “An introduction to intrusion-detection systems,” *Proc. Connect*, pp. 1–18, 2000.

[32] D. Anderson, T. Frivold, and A. Valdes, “Next-generation Intrusion Detection Expert System (NIDES): A summary,” *SRI Int.*, no. May 1995, p. 47, 1995.

[33] D. E. Denning, “An Intrusion-Detection Model,” *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 118–131, 1987.

[34] J. P. Anderson, “Computer security threat monitoring and surveillance,” 1980.

[35] P. A. Porras and A. Valdes, “Live Traffic Analysis of TCP/IP Gateways,” *Proc. 1998 ISOC Symp. Netw. Distrib. Syst. Secur. NDSS98*, 1998.

[36] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. September, pp. 1–58, 2009.

[37] M. Rich, “Evaluating Machine Learning Classifiers For Hybrid Network Intrusion Detection Systems,” Air Force Institute of Technology, 2017.

[38] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo, “Anomaly detection methods in wired networks: A survey and taxonomy,” *Comput. Commun.*, vol. 27, no. 16, pp. 1569–1584, 2004.

[39] E. Biermann, E. Cloete, and L. Venter, “A comparison of Intrusion Detection systems,” *Comput. Secur.*, vol. 20, no. 8, pp. 676–683, 2001.

[40] P.-L. D. Elike Hodo, Xavier Bellekens, Andrew Hamilton and C. T. and R. A. Ephraim Iorkyase, “Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System,” *2016 3rd Int. Symp. Networks, Comput. Commun.*, pp. 1–6, 2016.

[41] A. Jones and R. Sielken, “Computer system intrusion detection: A survey,” *Comput. Sci. Tech. Rep.*, pp. 1–25, 2000.

[42] J. Song, H. Takakura, Y. Okabe, and K. Nakao, “Toward a more practical unsupervised anomaly detection system,” *Inf. Sci. (Ny).*, vol. 231, pp. 4–14, 2013.

[43] M. Ahmed, A. Naser Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, 2016.

[44] H. Packard and E. Development, “HPE Security ArcSight Common Event Format Implementing ArcSight Common Event Format CEF).” Hewlett Packard Enterprise, p. 28, 2016.

[45] V. J. Hodge and J. Austin, “A Survey of Outlier Detection Methodoligies,” *Artif. Intell.*

Rev., vol. 22, no. 1969, pp. 85–126, 2004.

[46] F. E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[47] M. Goldstein and S. Uchida, “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data,” no. April, pp. 1–31, 2016.

[48] M. Agyemang, K. Barker, and R. Alhajj, “A comprehensive survey of numeric and symbolic outlier mining techniques,” *Intell. Data Anal.*, vol. 10, no. 6, pp. 521–538, 2006.

[49] R. J. Beckman and R. D. Cook, “Outlier … … …. s,” *Technometrics*, vol. 25, no. 2, pp. 119–149, May 1983.

[50] M. Markou and S. Singh, “Novelty detection: A review - Part 2:: Neural network based approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[51] R. Lippmann, “An Introduction to Computing with Neural Nets,” *IEEE ASSP Mag.*, vol. 4, no. 2, pp. 4–22, 1987.

[52] N. Japkowicz, C. Myers, and M. Gluck, “A novelty detection approach to classification,” *Proc. Fourteenth Int. Jt. Conf. Artif. Intell.*, pp. 518–523, 1995.

[53] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, and J. Ucles, “A hierarchical anomaly network intrusion detection system using neural network classification,” pp. 333–338, 2001.

[54] K. Labib and R. Vemuri, “NSOM: A real-time network-based intrusion detection system using self-organizing maps,” *Networks Secur.*, pp. 1–6, 2002.

[55] R. Smith, “Network-Based Intrusion Detection Using Neural Networks,” *New York*, pp. 1–18.

[56] S. Hawkins, H. He, G. Williams, and R. Baxter, “Outlier Detection Using Replicator Neural Networks,” in *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2000)*, 2000, pp. 170–180.

[57] C. Kruegel, G. Vigna, and G. Kruegel, Christopher and Vigna, “Anomaly detection of web-based attacks,” *Proc. 10th ACM Conf. Comput. Commun. Secur.*, vol. 3, no. November, p. 251, 2003.

[58] M. Ramadas, S. Ostermann, and B. Tjaden, “Detecting Anomalous Network Traffic with Self-organizing Maps,” *Recent Adv. Intrusion …*, vol. 2820, pp. 36–54, 2003.

[59] J. Ryan and M. Lin, “Intrusion Detection with Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 943–949, 1998.

[60] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, “Outlier Detection with Autoencoder

Ensembles," *SIAM Conf. Data Mining.*, pp. 90–98, 2017.

[61] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A neural network based database mining system for credit card fraud detection," in *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, 1997, pp. 220–226.

[62] S. Ghosh and D. L. Reilly, "Credit card fraud detection with a neural-network," in *System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on*, 1994, vol. 3, pp. 621–630.

[63] J. R. Dorronsoro, F. Ginel, C. Sgnchez, and C. S. Cruz, "Neural fraud detection in credit card operations," *IEEE Trans. Neural Networks*, vol. 8, no. 4, pp. 827–834, Jul. 1997.

[64] P. Barson, S. Field, N. Davey, G. McAskie, and R. Frank, "Detection of fraud in mobile phone networks," *Neural Netw. World*, vol. 6, no. 4, pp. 477–484, 1996.

[65] M. Taniguchi, M. Haft, J. Hollmén, and V. Tresp, "Fraud detection in communication networks using neural and probabilistic methods," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 1998, vol. 2, pp. 1241–1244.

[66] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 141, no. 4. p. 217, 1994.

[67] I. Diaz and J. Hollmén, "Residual generation and visualization for understanding novel process conditions," in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, 2002, vol. 3, pp. 2070–2075.

[68] T. Harris, "Neural network in machine health monitoring," *Prof. Eng.*, 1993.

[69] S. Jakubek and T. Strasser, "Fault-diagnosis using neural networks with ellipsoidal basis functions," in *American Control Conference, 2002. Proceedings of the 2002*, 2002, vol. 5, pp. 3846–3851.

[70] S. P. King, D. M. King, K. Astley, L. Tarassenko, P. Hayton, and S. Utete, "The use of novelty detection techniques for monitoring high-integrity plant," in *Control Applications, 2002. Proceedings of the 2002 International Conference on*, 2002, vol. 1, pp. 221–226.

[71] T. Petsche, A. Marcantonio, C. Darken, S. Hanson, G. Kuhn, and I. Santoso, "A neural network autoassociator for induction motor failure prediction," *Advances in Nural Information Processing Systems 8*, pp. 924-930, 1996.

[72] T. Verwoerd and R. Hunt, "Intrusion detection techniques and approaches," *Comput. Commun.*, vol. 25, no. 15, pp. 1356–1365, 2002.

[73] A. Ypma and R. P. W. Duin, "Novelty detection using Self-Organizing," *Fac. Appl. Physic*, 1996.

[74] J. M. Ko, Y. Q. Ni, J. Y. Wang, Z. G. Sun, and X. T. Zhou, “Studies of vibration-based damage detection of three cable-supported bridges in Hong Kong,” 2000.

[75] C. Surace and K. Worden, “Novelty detection in a changing environment: A negative selection approach,” *Mech. Syst. Signal Process.*, vol. 24, no. 4, pp. 1114–1128, 2010.

[76] K. Worden, “STRUCTURAL FAULT DETECTION USING A NOVELTY MEASURE,” *J. Sound Vib.*, vol. 201, no. 1, pp. 85–101, Mar. 1997.

[77] Y. Bengio and Y. Lecun, “Scaling Learning Algorithms towards AI,” *Large Scale Kernel Mach.*, no. 1, pp. 321–360, 2007.

[78] M. M. Moya and D. R. Hush, “Network Constraints and Multiobjective Optimization for One Class Classification,” *Neural Networks*, vol. 9, no. 3, pp. 463–474, 1996.

[79] S. Singh and M. Markou, “An approach to novelty detection applied to the classification of image regions,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 4, pp. 396–407, 2004.

[80] R. O. B. Saunders and J. S. Gero, “A curious design agent,” *Key Cent. Des. Comput. Cogn.*, pp. 345–350, 2001.

[81] G. C. Vasconcelos, M. C. Fairhurst, D. L. Bisset, and K. Ct, “A bootstrap-like rejection mechanism for multilayer perceptron networks,” in *II Simposio Brasileiro de Redes Neurais*, 1995, pp. 167–172.

[82] C. L. Wilson, J. L. Blue, and O. M. Omidvar, “Neural network auto-associator and method for induction motor monitoring,” in *World Congress on Neural Networks Proceedings II*, 1996, pp. 151–158.

[83] L. M. Manevitz, M. Yousef, N. Cristianini, J. Shawe-Taylor, and B. Williamson, “One-Class SVMs for Document Classification,” *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, 2001.

[84] S. Roberts and L. Tarassenko, “A Probabilistic Resource Allocating Network for Novelty Detection,” *Neural Comput.*, vol. 6, no. 2, pp. 270–284, 1994.

[85] P. Crook and G. Hayes, “A Robot Implementation of a Biologically Inspired Method for Novelty Detection,” *Proc. Towards Intelligent Mobile Roberts Conference*, 2001.

[86] J. Himberg, J. Ahola, E. Alhoniemi, J. Vesanto, and O. Simula, “The Self-Organizing Map as a tool in knowledge engineering,” *Pattern Recognit. soft Comput. Paradig.*, pp. 38–65, 2001.

[87] G. Williams and R. Baxter, “A comparative study of RNN for outlier detection in data mining,” *IEEE Int. Conf. Data Min.*, no. December 2002, pp. 1–16, 2002.

[88] M. G. Dondo, N. Japkowicz, and R. Smith, “AutoCorrel: a neural network event correlation approach,” *Data Mining, Intrustion Detect. Inf. Assur. Data Networks Secur.*,

vol. 6241, April, p. 62410N, 2006.

[89] M. G. Dondo, P. Mason, N. Japkowicz, and R. Smith, “AutoCorrel II: a neural network event correlation approach.,” *Data Mining, Intrusion Detect. Inf. Assur. Data Networks Secur.*, vol. 6570, April, p. 65700H, 2007.

[90] M. Dondo, “Investigation of a Neural Network Implementation of a TCP Packet Anomaly Detection System,” *Defense R&D Canada Technical Memorandum*, no. 2004-208, May, 2004.

[91] Mukhopadhyay Indraneel and Chakraborty Mohuya, “Artificial Neural network Modeling of Intrusion Detection & Prevention System,” *IEM Int. J. Manag. Technol.*, vol. 2, no. 1, pp. 107–110, 2012.

[92] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, “Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams,” *AICS - Artif. Intell. Cyber Secur.*, no. 2012, pp. 224–231, 2017.

[93] I. Mukhopadhyay and M. Chakraborty, “Hardware Realization of Artificial Neural Network Based Intrusion Detection &amp;amp; Prevention System,” *J. Inf. Secur.*, vol. 5, no. 4, pp. 154–165, 2014.

[94] A. K. Jain and J. Mao, “Artificial Neural Network: A Tutorial,” *Communications*, vol. 29, pp. 31–44, 1996.

[95] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[96] B. A. Olshausen and D. J. Field, “How Close Are We to Understanding V1?,” *Neural Comput.*, vol. 17, no. 8, pp. 1665–1699, 2005.

[97] L. von Melchner, S. L. Pallas, and M. Sur, “Visual behavior mediated by retinal projections directed to the autitoriy pathway,” vol. 404, no. 6780, pp. 871–876, 2000.

[98] B. Widrow and M. A. Lehr, “30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation,” *Proc. of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[99] Q. Jarosz, “File:Neuron Hand-tuned.svg,” *Wikimedia Commons*, 2009. [Online]. Available: https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg. [Accessed: 14-Dec-2017].

[100] P. Winston, “6.034 Artificial Intelligence,” *Massachusetts Institute of Technology: MIT OpenCourseWare*, 2010. [Online]. Available: https://ocw.mit.edu.

[101] A. Castrounis, “Artificial Intelligence, Deep Learning, and Neural Networks Explained,” *InnoArchiTech*, 2016. [Online]. Available: https://www.innoarchitech.com/artificial-intelligence-deep-learning-neural-networks-explained/?utm_source=kdnuggets&utm_medium=post&utm_content=postlink&utm_campaign=republish. [Accessed: 09-Jan-2018].

[102] A. Ng, *CS 294A Sparse autoencoder Lecture Notes*. Stanford, CA: Stanford University, 2011.

[103] D. Tanikic and V. Despotovic, "Artificial Intelligence Techniques for Modelling of Temperature in the Metal Cutting Process," in *Metallurgy - Advances in Materials and Processes*, InTech, 2012.

[104] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," *Proc. Int. Jt. Conf. Neural Networks*, vol. 1, pp. 593–605, 1989.

[105] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[106] K. Bauer, *OPER 685 (Applied Multivariate Analysis) Course Notes*. Wright-Patterson AFB, OH: Air Force Institute of Technology, 2016.

[107] G. E. Hinton, "Connectionist learning procedures," *Artif. Intell.*, vol. 40, no. 1–3, pp. 185–234, Sep. 1989.

[108] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," San Diego, 1985.

[109] A. Candel, E. LeDell, V. Parmar, and A. Arora, *Deep Learning With H2O*, Fifth., March. Mountain View, CA: H2O.ai, Inc., 2016.

[110] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[111] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993.

[112] M. Claesen and B. De Moor, "Hyperparameter Search in Machine Learning," arXiv:1502.02127v2, Apr. 2015.

[113] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 437–478, 2012.

[114] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[115] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," *Adv. Neural Inf. Process. Syst.*, vol. 19, no. 1, p. 153, 2007.

[116] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training," *Aistats*, vol. 5, pp. 153–160, 2009.

[117] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” *Proc. 26th Annu. Int. Conf. Mach. Learn. - ICML ’09*, vol. 2, no. 1, pp. 1–8, 2009.

[118] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, P. Vincent, A. Courville, and J. Bergstra, “Unsupervised and Transfer Learning Challenge: a Deep Learning approach,” *JMLR W& CP Proc. Unsupervised Transf. Learn. Chall. Work.*, vol. 27, pp. 97–110, 2012.

[119] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” *Neural Networks*, vol. 32, pp. 333–338, 2012.

[120] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[121] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, “Pedestrian detection with unsupervised multi-stage feature learning,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3626–3633, 2013.

[122] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Indoor Semantic Segmentation using depth information,” arXiv:1301.3572v2, Mar. 2013.

[123] S. E. Kanou *et al.*, “Combining modality specific deep neural networks for emotion recognition in video,” *Proc. 15th ACM Int. Conf. multimodal Interact. - ICMI ’13*, pp. 543–550, 2013.

[124] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, “Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks,” *CoRR*, vol. abs/1312.6, pp. 1–13, 2013.

[125] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 1–9, 2015.

[126] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[127] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[128] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” 2012.

[129] B. Singh, S. De, Y. Zhang, T. Goldstein, and G. Taylor, “Layer-Specific Adaptive Learning Rates for Deep Networks,” arXiv:1510.04609v2, vol. 2, no. 1, Oct. 2015.

[130] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. M??ller, “Efficient backprop,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 9–48, 2012.

[131] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, no. 2010, pp. 8609–8613, 2013.

[132] D. R. Wilson and T. R. Martinez, “The general inefficiency of batch training for gradient descent learning,” *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, 2003.

[133] M. A. Kramer, “Autoassociative neural networks,” *Comput. Chem. Eng.*, vol. 16, no. 4, pp. 313–328, 1992.

[134] G. E. Hinton, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[135] J. Bergstra, D. Yamins, and D. D. Cox, “Making a Science of Model Search,” avXir:1209.5111v1, vol. 1, pp. 1–11, Sep. 2012.

[136] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[137] P. Koch, B. Wujek, O. Golovidov, and S. Gardner, “Automated Hyperparameter Tuning for Effective Machine Learning,” SAS514-2017, pp. 1–23, 2017.

[138] S. Yang and G. Lee, “Neural network design by using Taguchi method,” *J. Dyn. Syst. Meas. Control*, vol. 28, no. 1, pp. 6–9, 1999.

[139] J. F. C. Khaw, B. S. Lim, and L. E. N. Lim, “Optimal design of neural networks using the Taguchi method,” *Neurocomputing*, vol. 7, no. 3, pp. 225–245, 1995.

[140] G. E. Peterson, W. E. Bond, D. C. Clair, and S. R. S. Aylward, “Using Taguchi’s Method of Experimental Design to Control Errors in Layered Perceptrons,” *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 949–961, 1995.

[141] Y. S. Kim and B. J. Yum, “Robust design of multilayer feedforward neural networks: An experimental approach,” *Eng. Appl. Artif. Intell.*, vol. 17, no. 3, pp. 249–263, 2004.

[142] F. Sánchez Lasheras, J. A. Vilán Vilán, P. J. García Nieto, and J. J. del Coz Díaz, “The use of design of experiments to improve a neural network model in order to predict the thickness of the chromium layer in a hard chromium plating process,” *Math. Comput. Model.*, vol. 52, no. 7–8, pp. 1169–1176, 2010.

[143] M. S. Packianather, P. R. Drake, and H. Rowlands, “Optimizing the parameters of multilayered feedforward neural networks through Taguchi design of experiments,” *Qual. Reliab. Eng. Int.*, vol. 16, no. 6, pp. 461–473, 2000.

[144] F. J. Pontes, G. F. Amorim, P. P. Balestrassi, A. P. Paiva, and J. R. Ferreira, “Design of experiments and focused grid search for neural network parameter optimization,” *Neurocomputing*, vol. 186, pp. 22–34, 2016.

[145] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Cambridge, Mass: MIT Press, 2010.

[146] B. Wujek, P. Hall, and F. Güneș, "Best Practices for Machine Learning Applications," SAS2360-2016, pp. 1–23, 2016.

[147] D. C. Montgomery, *Design and Analysis of Experiments*, 8th ed. Hoboken, NJ: Wiley, 2012.

[148] D. C. Montgomery, E. Peck, and G. Vining, *Introduction to Linear Regression Analysis*, 5th ed. Wiley, 2012.

[149] R Core Team, "R: A Language and Environment for Statistical Computing." R Foundation for Statistical Computing, Vienna, Austria, 2017.

[150] RStudio Team, "RStudio: Integrated Development Environment for R." RStudio, Inc., Boston, MA, 2017.

[151] "JMP Pro." SAS Institute Inc., Cary, NC, 2017.

[152] The MathWorks Inc, "MATLAB Statistics and Machine Learning Toolbox Release 2016b." Natick, MA, 2016.

[153] H. Wickham, "tidyverse: Easily Install and Load 'Tidyverse' Packages." 2017.

[154] R. Robertson, F. Tran, J. Mejia, and C. Mourani, "IPtoCountry: Convert IP Addresses to Country Names or Full Location with Geoplotting." 2016.

[155] H. Bengtsson, "R.utils: Various Programming Utilities." *R package version 2.6.0,* 2017 [Online]. Available: https://cran.r-project.org/package=R.utils.

[156] M. Kuhn, "caret: Classification and Regression Training." *R package version 6.0-78*, 2017. [Online]. Available: https://cran.r-project.org/package=caret.

[157] H. Wickham, "scales: Scale Functions for Visualization." *R package version 0.5.0,* 2017. [Online]. Available: https://cran.r-project.org/package=scales.

[158] The H2O.ai Team, "h2o: R Interface for H2O." *R package version 3.16.0.2,* 2017. [Online]. Available: https://cran.r-project.org/package=h2o.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)* 22-03-2018	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From - To)* Aug 2016 - March 2018

4. TITLE AND SUBTITLE Cyber Data Anomaly Detection Using Autoencoder Neural Networks	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Butt, Spencer A. Captain, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENS-MS-18-M-113
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Cyber Command LTC Cade M. Saie, PhD 8825 Beulah St For Belvoir, VA 22060 cade.m.saie.mil@mail.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** ARCYBER **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The Department of Defense requires a secure presence in the cyber domain to successfully execute its stated mission of deterring war and protecting the security of the United States. With potentially millions of logged network events occurring on defended networks daily, a limited staff of cyber analysts require the capability to identify novel network actions for security adjudication. The detection methodology proposed uses an autoencoder neural network optimized via design of experiments for the identification of anomalous network events. Once trained, each logged network event is analyzed by the neural network and assigned an outlier score. The network events with the largest outlier scores are anomalous and worthy of further review by cyber analysts. This neural network approach can operate in conjunction with alternate tools for outlier detection, enhancing the overall anomaly detection capability of cyber analysts.

15. SUBJECT TERMS

Cyber anomaly detection, Intrusion detection, Neural networks, Autoencoder, Anomaly detection, Intrusion detection, Intrusion protection

16. SECURITY CLASSIFICATION OF: 1.			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Raymond Hill
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	151	**19b. TELEPHONE NUMBER** *(include area code)* (937)255-3636 x7469 raymond.hill.2@us.af.mil