FACULTY OF SCIENCE & TECHNOLOGY

BSc (Hons) Forensic Computing and Security

May 2019

A comparison of neural network-based network IDS techniques for achieving interpretability

by

Samuel William Denton

Faculty of Science & Technology

Department of Computing and Informatics

Final Year Project

Abstract

Deep learning and neural network techniques have proven to be an effective solution for network intrusion detection, and look set to dominate the field of cyber security. However, these technologies suffer a major drawback, their black box nature. The inability to interpret, explain and understand how neural networks reach classification results hinders our ability to trust and rely on them for important tasks such as intrusion detection, especially when applied to critical infrastructure. Lack of understanding makes it hard to be sure that deep learning and neural network techniques will transition reliably from controlled testing environments to working situations, where the results from their classifications will mean the difference between an attack being detected or allowed into a network, with potentially catastrophic consequences.

Because of this, the field of explainable AI (XAI, interpretable AI) has been in development almost as long as neural networks have existed. Many techniques for interpreting and explaining the workings of the neural network, look at how they consider input features, how they weigh different aspects of the data, how well they understand the context of the data and why a classification was made.

In this paper I will examine some of the current 'state of the art' techniques which can improve on interpretability and explainability when applied to network intrusion detection systems, looking at how they affect the systems accuracy and false positive rates, as well as giving detailed explanations as to how they aid our understanding of the systems. I will show how each method can help develop trust in the neural network, and how a better understanding can help build more robust, reliable IDSs. I will also give insight into how these methods can be improved and combined in future works to further improve the field of XAI and IDS.

Dissertation Declaration

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed: _____.

Name: Samuel William Denton

Date: 30/05/2019

Programme: BSc (Hons) Forensic Computing and Security

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

Signed:  .

Name: Samuel William Denton

Date: 30/05/19

Acknowledgments

I would like to thank Neetesh Saxena, my supervisor for this project, for his helpful advice and input; Neil Gallivan for his support and advice on dealing with difficult situations; as well as Nikki Denton, Brian Denton and Margaret Hadley for their support in helping me stay motivated and on track.

TABLE OF CONTENTS

Contents

LIST OF FIGURES

1 INTRODUCTION

Recently, a greater number of intrusion detection systems, especially network IDSs are turning to machine learning techniques like deep learning and neural networks to power their systems. They have been shown to be highly effective at detecting a wide range of potential attacks and are outperforming most classical algorithmic methods. A number of neural network driven network IDSs are now available and widely used across a number of industries, including critical infrastructure. As these technologies propagate further into modern society we will rely more and more on their results, and it is critical that we understand every aspect of how they operate, so we can predict how they will react during edge cases and unknown situations. However, as with all deep learning and neural network applications there is a major drawback, the Black Box nature of neural networks.

1.1 PROBLEM DEFINITION

Since the first neural networks were proposed in 1943, it has been apparent that, while this technology has the potential to solve problems beyond the scope of a human brain and standard computation algorithms, we struggle to explain and interpret how the network reached the solution. This is an issue because many of the problems these networks are tackling could have significant impact on the human population, and to be able to trust the results, we need to understand why they are being made. Being able to understand the results also allows us to correct mistakes or misclassifications and build more powerful and reliable neural networks. If the neural network could explain its classifications and hence detections in comprehensible natural language, we would be able to quickly understand what caused the detection, and if additional action is required. This is invaluable, both to generate greater levels of trust in the system, but also identify false positives, of which IDSs tend to show many. A quote from the paper 'Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)' (Adadi, A. and Berrada, M. 2018) talks about the limitations of using black-box systems, despite their promise:

> *'even with such unprecedented advancements, a key impediment to the use of AI-based systems is that they often lack transparency. Indeed, the black-box nature of these systems allows powerful predictions, but it cannot be directly explained*'.

This sums up well the current state of almost all systems utilising machine learning techniques like neural networks and shows the need for greater levels of interpretability in our future systems. Because of the desire to understand neural networks, the field of explainable artificial intelligence (interpretable AI, XAI) is almost as old as AI itself. Not much progress was made in the field until the last 10 years, but it is becoming apparent that for AI to progress further into modern society, we must learn to trust it, and the first step to trust is understanding. Because of this, there has been a big push for XAI and many techniques have been developed. The Defence Advanced Research

Projects Agency (DARPA) has expressed great interest in the field, and in 2016 created a program specifically for aiding and funding the progression of XAI. Quoting their official website:

> *"the effectiveness of these systems is limited by the machine's current inability to explain their decisions and actions to human users. Explainable AI—especially explainable machine learning—will be essential to understand, appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners"* (Gunning, D. 2016).

1.2 IMPACT

This problem is especially relevant when neural networks are applied to network intrusion detection systems, and even more so when we wish to apply them to critical infrastructure. These systems are the backbone for many important aspects of modern civilisation such as banking, communication, medical care and many others and as such it is imperative that we understand and trust the systems protecting them (Amarasinghe, K. and Manic, M. 2018) (Amarasinghe, K., Kenney, K. and Manic, M. 2018).

Regardless of how accurate and robust the systems become, there will always be edge cases which have not been tested against. If an attack contained data which the network had never seen before, we have no way of ensuring it would react in a suitable manner, since we don't understand how it evaluates the data. This could have catastrophic affects as an attack on critical infrastructure which is not detected could cause irreparable damage, and potentially be life threatening. Less dangerous, but maybe more likely is the opposite. If normal but unusual data is shown as an attack, time and money could be wasted trying to prevent something which does not exist.

1.3 CHALLENGES

This is an especially difficult field of research, and many researchers doubt it is even possible to create fully explainable AI. For this reason, explainability and interpretability have been given separate definitions in recent years. While interpretability is far more achievable, it is by no means easy. It requires an in-depth knowledge of how the network evaluates input data, and requires analysis of both its internal structure, learning methods and biases. There are a number of methods to achieve this by manually examining the weights of connections in a neural network, however these give limited information, take a long time and are inappropriate for larger networks, such as would be used for an IDS.

1.4 THE POTENTIAL OF XAI

If interpretability can be achieved, the payoff is great; being able to fully predict the results of the network allows us to understand how the system will react in different situations, allow us to analyse what went wrong during misclassifications and improve the overall accuracy of the system. It can potentially prevent disastrous situations by understanding how edge cases might be

handled, and prevent time being wasted on false positives by showing exactly what triggered the detection. (Došilović, F., Brčić, M. and Hlupić, N. 2018) (Hagras, H. 2018)

Explainability would have an even greater benefit, as it would allow for the network to explain its reasoning for classifications and could help us better understand the extent to which the network understands the context of the input data and can help us remove biases which may be gained during training.

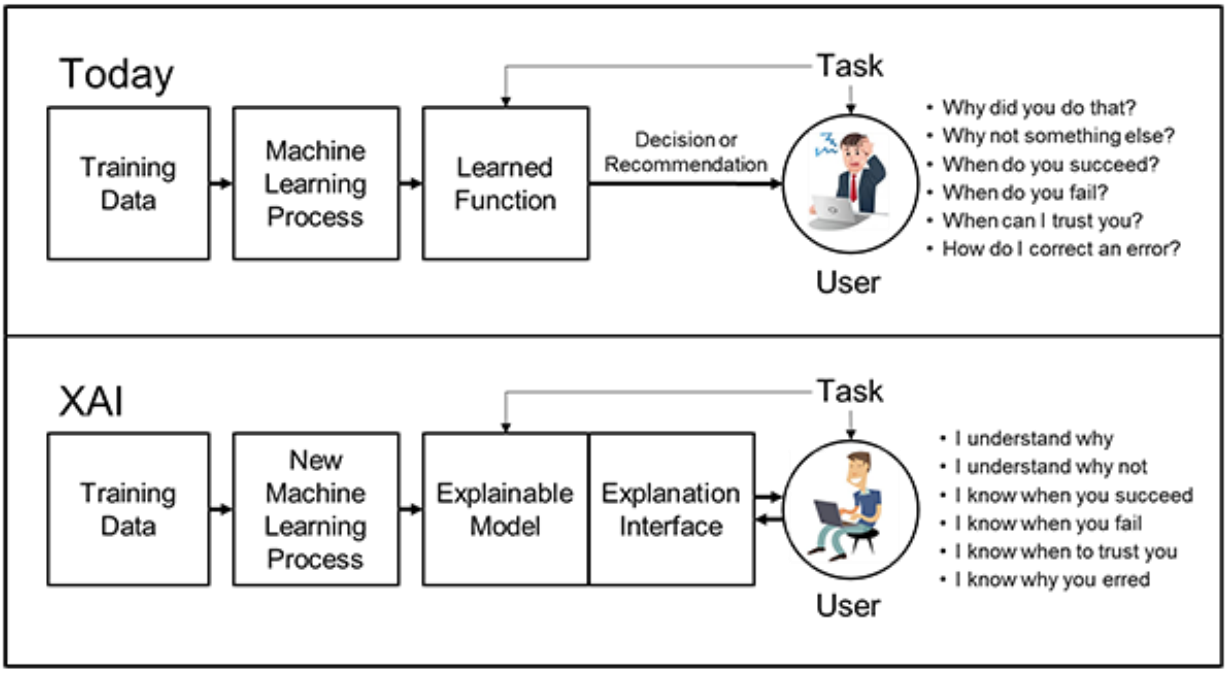


Figure 1: The basic principal behind the work process of AI vs XAI (Gunning, D. 2016)

1.5 AIMS:

To research the 'state of the art' in solutions for the black box issue within Artificial Intelligence driven Network Intrusion Detection Systems, determine their suitability and recommend the most fitting solutions from current proposed designs, with potential improvements.

1.6 OBJECTIVES:

1. Explore current proposed solutions to the black box problem in neural network-based network intrusion detection systems
2. Determine the positives and negatives of each method, pertaining to how they affect detection rates and how well they fit my definitions for interpretability and explainability
3. Evaluate solutions to determine best fit to solve the problem, give recommendation for which solution is best and provide insight into potential improvement

1.7 RISK ANALYSIS

One risk for this project is the rapid pace of progression in both artificial intelligence technologies and the field of security. In the months it takes to complete this work many new advancements are predicted to occur, and AI in particular has always exceeded expectations in its rapid developments. These could potentially make this work obsolete, as the methods I describe could conceivably be outdated by the time I am complete. I can mitigate this risk by including in my research, time for keeping up to date on the latest developments and either adapting my work or using it as evidence in the projects favour.

One risk that affects many AI projects is that classical computing techniques and algorithms might be considered good enough, at least for now, following the mentality of “If it ain't broke, don't fix it”. This seems unlikely since many intrusion detection systems have already adopted machine learning techniques, however advancements in classical computing methods could outpace AI development. This does not greatly affect the progress of my work, as interpretability will be useful for any future projects which choose to use neural networks.

Another risk is that interpretability may not be considered important enough to devote additional research to, and efforts could be diverted into improving the accuracy, speed and generalisation of the neural networks. This is a financial driven risk as business may care more about the IDS being robust and accurate than being able to explain its actions.

From a personal health standpoint, spending upwards of eight hours a day working on a computer, especially a single project, can lead to back and eye strain, and can cause issues with sleeping. This can be mitigated by using a comfortable, upright seating position, taking regular breaks away from the screen and exercising regularly. I will also be wearing glasses which block harsh blue light from monitors which can aid in reducing eye strain and can help avoid sleep disruption.

1.8 SUMMARY

In this paper I will show some of the current leading techniques for improving interpretability and explainability, how they can be applied to IDSs, and give detailed explanations as to how they can improve both the field of deep learning and cyber security. I will also put emphasis on ensuring the techniques examined do not impede the accuracy of the system, and as such will include figures produced by testing preliminary models utilising the proposed techniques. The purpose of this work is to aid in the development of future IDSs which have an emphasis on interpretability and explainability. It is intended to show which method or methods show most promise in developing this field, and direct further research towards these techniques.

2 BACKGROUND STUDY

In this section, I will provide a comprehensive background study of both intrusion detection and artificial intelligence, focussing on network intrusion detection and neural networks, with all the information required to understand the content discussed in both my work, and my reference material. Any keyword definitions and additional information will be presented in Appendix A.
I will then present the five methods for improving interpretability which I have examined, with positives and negatives of each, as well as details of their architecture and implementation.

2.1 INTRUSION DETECTION SYSTEMS

Intrusion detection systems fall into two categories, Active and passive intrusion detection. Active IDSs, also known as intrusion detection and prevention systems, are configured to automatically block suspicious attacks without intervention from a human. Passive IDSs focus only on detecting and reporting suspicious activity for a human expert to act upon. While the solutions examined in this paper are all passive IDSs, each could have a preventative system added if required.
IDSs can further be broken down into one of four categories, which focus on different areas of a computer system.

Network intrusion detection systems operate on a separate network appliance, and are connected to some part of the network, often at a switch. They monitor all traffic passing through that part of the network and analyse the data for several different attacks, including DoS and probe attacks.

Host based intrusion detection systems operate in a similar manner to network IDSs, except they are installed on a single workstation or server. They generally look for specific attacks on critical servers and are used in parallel with network IDSs. Their main drawback is that they must be maintained individually on each machine where they are installed.

Knowledge based intrusion detection systems also known as signature based IDSs reference a database of previously known attacks and vulnerabilities. They are highly effective against known attacks but are ineffective against new attacks. This means they must be updated regularly with the latest attacks and exploits. They should be used in conjunction with other IDSs which can detect unknown attacks.

Behaviour based intrusion detection systems also known as anomaly IDSs learn the normal data patterns on the system they are installed on and flag any deviations from this pattern. They can be effective against many types of attack however they tend to have high false positive rates due to limited knowledge of the context of the data they analyse. Anything from visiting an unknown website to connecting smart devices could trigger a false positive.

This paper will focus on network intrusion detection systems. There are many types of attack which are normally detected by network IDSs, and the datasets used to compare the proposed methods contain four attack types, DoS, Probe, Remote to Local and User to Root, however, since some of these methods were only tested on a subset of the dataset containing only DoS and Probe attacks, I will focus on these.

Denial of Service attacks (DoS) are a method for disrupting performance on a target network. The premise of the attack is to flood the network with huge amounts of random data packets. If the number of packets is greater than or close to the maximum bandwidth of the network, data packets sent and received by the network can be delayed or lost. This results in significant loss of performance and can cause complete loss of service. There are many ways to create a DoS attack including; Distributed Denial of Service (DDoS) where a large number of computers, often part of a botnet, are directed to send data to the same target; Amplified Denial of Service, where an attacker sends queries to a DNS resolver or time server using the targets IP address, so the results of the queries floods the target network; and many others.

DoS attacks can be detected by looking at incoming data and determining its source IP address. If a large quantity of data is being received from a single IP address, there is a good chance it is a DoS attack. This becomes more challenging if other types of DoS attacks are used, as the data can be sent from many different addresses; however, these attacks still leave detectable patterns in the data which an IDS can use to identify the attack.

DoS attacks can cause significant reduction in network performance, or even cause complete failure. If the target is a webserver there is a strong likelihood the server will become unavailable during the attack.

DoS attacks can be prevented by blacklisting the IP address or addresses sending the data. This tells the network to ignore incoming data from that address. This method can be impractical for some types of DoS attacks; hence many webserver hosts use a Content Delivery Network (CDN) service which monitors incoming data for them and routs positive data to the server while blocking data which appears to be an attack. This works as the servers used by the CDN service have a huge bandwidth and are unaffected by all but the largest DoS attacks.

Probe attacks are used to scout the defences in place on a network. Probing normally comes in three parts; port scanning, enumerating and vulnerability assessment. These can give an attacker a better idea of where the weaknesses might be in a network, and how to structure their attacks in future.

Probe attacks can be very difficult to detect as they can come in many forms, however many methods using machine learning have been implemented in IDS packages to learn and recognise patterns in incoming and outgoing traffic on a network to indicate a potential probe. The difficulty in detecting these attacks is due to the fact that much of the information requested by probe attacks

is also needed for programs and features on the network to function. This makes it difficult to hide the information without interrupting services on the network.

Probe attacks can be devastating to a network if a vulnerability is found, as it can make it easy for an attacker to gain access. This is why penetration testing and vulnerability scanning is important as it can show potential issues which an attacker could exploit.

Probe attacks can't be entirely prevented, but there are measures you can take to reduce the chances of vulnerabilities being discovered. Closing unused ports, keeping services that require open ports up-to-date and scanning open ports and log files can help detect when a probe attack is happening, and an intrusion prevention system (IPS) can monitor these points of access and attempt to prevent them, either by temporarily blocking a port, dropping packets to a specific address or service, or resetting a connection. Unfortunately, how an IPS reacts to a probe attack also tells an attacker about the network, and further probes may take this into account.

2.2 Artificial Intelligence

2.2.1 Artificial Intelligence

Artificial Intelligence is a broad term referring to a large research area covering any computer program capable of simulating intelligent behaviours or human reasoning. There are two main categories of AI, weak and strong. Weak AI is a tool focussing on a specific task, enhanced with the ability to learn or reason about its subject. All AI currently in existence is weak AI. Strong AI, also known as Artificial General Intelligence (AGI), is the main goal for many larger research institutes. Strong AI will have the capacity to generalise its reasoning to any task presented to it, and will match or exceed humans in its understanding, reasoning and learning capacities.

2.2.2 Machine Learning

Machine Learning is one application of AI which provides systems with the ability to automatically learn and improve from 'experience'. This means it is able to change some part of how it operates to improve accuracy on a task, based on past performance. They generally operate by receiving a large amount of training data and after running that data through its operations, receiving some input, either from a human or algorithm, to determine how well it did. It can use this repeated feedback to alter parts of its internal structure to attempt to improve on the result. There are 4 main types of machine learning.

Supervised Learning takes labelled data as training data and is told exactly how far from a correct result it achieved. These labels are created using human knowledge. This is used for problems where we wish to classify data which we already understand well and know how it should be classified before-hand. We use AI for these problems to help with scale, speed and accuracy.

Unsupervised Learning uses un-labelled data. It allows the program to look for patterns in the data that were not explicitly programmed by a human, meaning it is allowed to create its own classifications or labels. This makes it a useful tool for finding patterns that may be too obscure for a human, or patterns that appear in higher dimensions of data which are difficult for a human to comprehend. We can use unsupervised learning to find classifications in data which would require too much time, complexity or understanding with traditional computing methods.

Semi-supervised Learning methods train using a combination of labelled and un-labelled data. We use this method when we have access to labelled data but collecting and classifying the data before training takes too long using traditional methods. It allows us to train a model using the small amount of labelled data available, alongside the large amount of un-labelled data which is yet to be classified.

Reinforcement Learning is a method were the program interacts with an environment, normally virtual, to discover what effects its actions have. It learns by being assigned a 'fitness score' based on how it performed and calculating what changes it can make to its internal structure to improve that fitness. This method is commonly used to train AI to play games, where progress through the game is its fitness, and optimising input controls allows it to progress further. Training AI to play games is normally a way to show off the 'state of the art' however, this method can also be used for a huge range of applications and might be the most diverse method of machine learning. Recommendation engines for social media, traffic light controls, medication development and robotics are all examples of where reinforcement learning can be beneficial.

2.2.3 Deep Learning

Deep Learning is a technology which emulates the learning methods used by humans. It learns using algorithms stacked in a hierarchy of increasing complexity and abstraction to develop understanding of a subject over time. Each algorithm in the hierarchy applies some non-linier transformation on its input to create a statistical model as an output. These transformations mutate over time to increase the accuracy of the model over many iterations until an acceptable level of accuracy is achieved. Deep Learning can utilise any of the machine learning methods.

2.2.4 Neural Networks

A Neural Network is a type of Deep Learning technology which normally focuses on pattern recognition in higher dimensional data. Neural Networks are structured with a number of layers. Each layer has a number of cells (aka nodes, neurons), and each cell is connected to cells from other layers, although there are many different ways to connect the cells based on the 'Architecture' type used. The connections between cells are given a 'weight', and each cell has an 'activation function'. The weight is what changes as the network learns. The first layer is the input. This layer receives raw data from the dataset and has the same number of inputs as there are

variables in the data. For example, if the input was a set of images, each pixel in the image would correspond to an input cell, with a value describing the pixel. For images of 28*28 pixels, this equates to 784 input cells. The next 'N' number of layers is the hidden layers. The number of hidden layers and number of cells per layer can be any number depending on the complexity of the problem. Larger networks can find more complex patterns but take longer to train and are more prone to 'overfitting'. The final layer is the output. The number of cells in the output equates to the number of options the network can pick between. In our image example above, if I wanted the neural network to tell me if the image was of a dog or a cat, there would be two output cells, one representing dog and one for cat. The classification result is whichever the network gives a higher value. If we wanted our network to drive a car, there would be an output cell for each control of the car; acceleration, turning angle, breaking, etc. The output of the network is the sum of each connection weight, after it has been passed through the activation function of the cell following it. Each layer has an additional 'bias cell' as an input, which is always set to 1, with its own connection weight, to correct for situations where the inputs to a layer are all 0. There are a huge number of different configurations or architectures that have been developed and each is most suitable for a different type of problem.

2.2.5 Structure of a Neural Network

There is huge variety in the ways a neural network can be structured, called its architecture. Different architectures are suitable for different applications and is one of the main considerations when building a neural network. This can be one of the main factors in the accuracy of results produced, and is subject to heavy debate, especially when new architecture types utilising new types of cells are being developed all the time. Figure 2 shows some of the most common cell types and architectures as of 2015. Most of the architectures with hidden layers are scalable to have any number of hidden layers and cells per layer. I will describe some of the differences between the most important aspects of the most used types.

The **Perceptron** is the most basic neural network possible. It takes 2 inputs, sums the weights of the connections and applies an activation function to give a result. Any problem which is mathematically linearly separable can be solved with a perceptron. They are mainly used to in education to show the structure of a cell within a network.

Feed Forward Neural Networks (FFNN) is any neural network using no loops, meaning the values can only be passed forward through the network. Types of FFNN include Multi-Layer perceptron's (MLP) (shown in figure 2 as Feed Forward), Radical Basis Network (RBF) and Deep Feed Forward (DFF).

MLP's are defined as having only one hidden layer and using a logistic function such as sigmoid as its activation function. Logistic functions produce an output between two values, i.e. -1, 1 in the case of sigmoid, making them useful for classification problems.

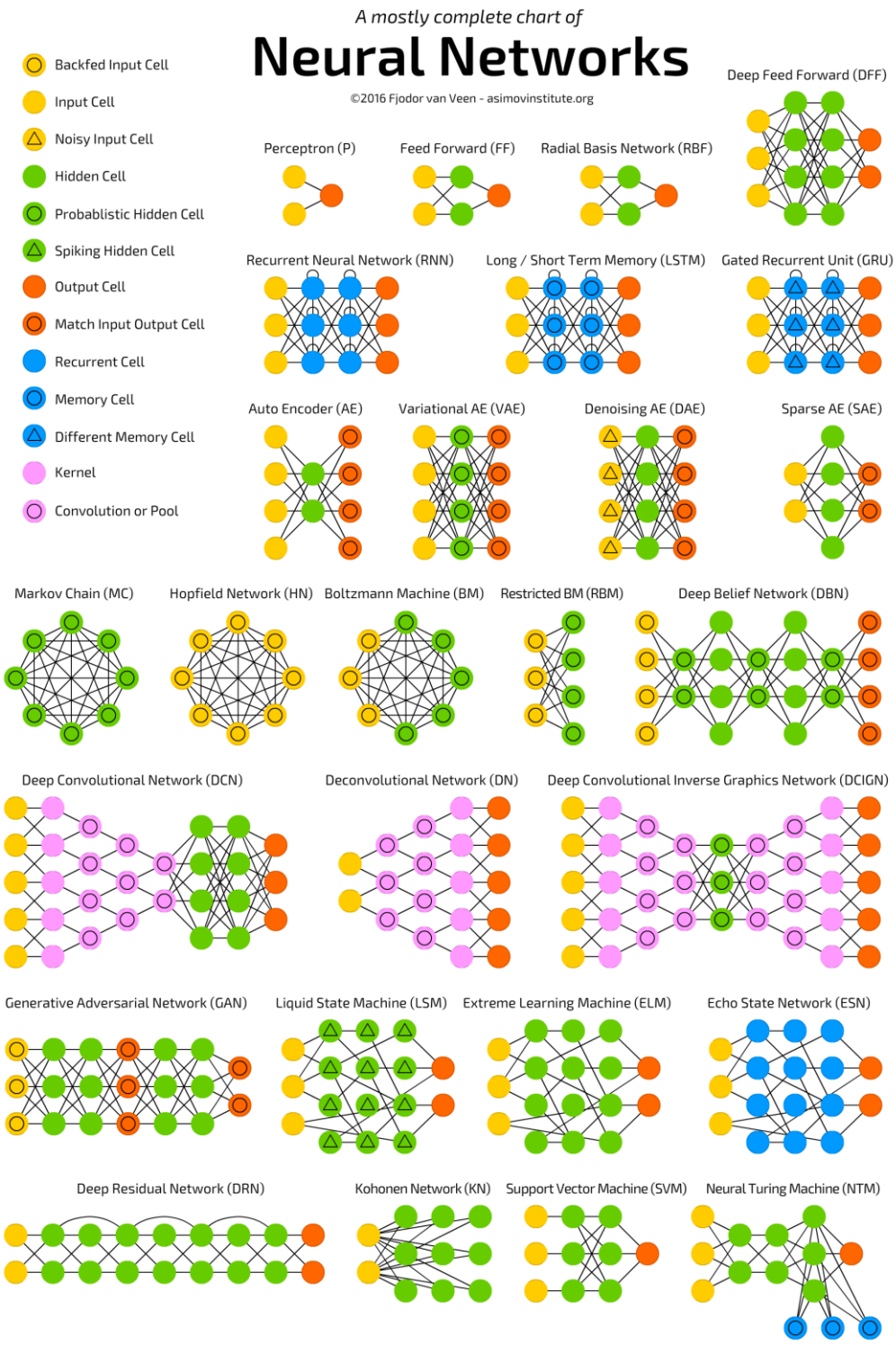


Figure 2: A mostly complete chart of Neural Networks (VAN VEEN, F. 2016)

RBF's have the same structure as an MLP but use a radical bias function as its activation function. These function types give an approximation as to how far the current value is from a target, making them useful for approximation problems.

DFF networks are essentially feed forward networks with multiple hidden layers. Having multiple layers allows the network to work with more complex data sets, involving many dimensions. For example, a FFNN can solve problems which are separable in two-dimensional space, such as XOR operations, but would struggle to gain an understanding of data with many 'features' such as classifying an image as either dog or cat. The higher dimensionality refers to the fact that there are many features that make up the definition of a dog vs cat, while data to be classified with an XOR operation only has two features.

The other types of neural network gain complexity and functionality using different cell types and connection structures. I will describe the different cell types, additional information about activation functions as well as definitions for keywords which might appear throughout my work and the material I have referenced in Appendix A**.**

2.3 EXPLAINABILITY VS INTERPRETABILITY

To fully appreciate the differences between solutions examined in this paper, as well as what I am looking for in an ideal solution, it must first be clear what the definitions of interpretability and explainability are, and the differences between them (Choudhury, A. 2019) (Došilović, F., Brčić, M. and Hlupić, N. 2018) (Gall, R. 2018).

2.3.1 Interpretability

Interpretability refers to the extent to which we can describe the cause effect relationship between the input data and output result. It has also been described as our ability to predict the output of the neural network based on changes to the input.

2.3.2 Explainability

Explainability expands on this by showing the extent to which we understand the internal mechanics of a neural network and can describe the networks reasoning behind classifications. This does not simply mean showing the layout of nodes and connections but means being able to extrapolate deeper meaning and learned knowledge from the network.

The difference is subtle, but explainability requires a much deeper level of comprehension. For example, an interpretable neural network might be able to tell you which features of a dataset it considers to be important, while an explainable neural network could tell you why those features are important, and answer questions about the process leading to the development of that knowledge.

In an ideal solution this might present itself in a few ways. An interpretable neural network might be able to produce a visual or natural language representation of which input features it considers to be important for a classification. It might be able to describe how changing different input values

would change the classification. It could give examples of what input value ranges would result in a specific classification. An explainable neural network should be able to take this a step further and be able to show an ability to answer 'why' questions. Why is this input feature important? Why did changing this input have the effect that it did? Once these questions can be answered, we can begin to assume it is an explainable neural network.

2.4 Current Network IDSs

Almost all modern intrusion detection systems employ the use of machine learning and neural network techniques. They have been shown to consistently outperform classical computing methods in both detection rates and false positives. However, none of the currently available products have looked towards improving interpretability or explainability, instead focusing on making the systems as accurate as possible. Some examples of this are Darktrace and FireEye which are network-based IDSs utilising neural networks for their core detection algorithm. These tools have been heavily adopted due to their high detection rates and are used in numerous critical infrastructures. However, these solutions have been reported to have high false positive rates, and it is often unclear why these normal data were classified as an attack (Francis, R. 2017) (Turnbull, M. 2018).

2.5 KDD Training and Testing Datasets

Each of the solutions I am examining in this paper were tested using the KDD datasets, either KDD cup 99, or its replacement NSL-KDD. While there are a few differences between the datasets, they share most of their fundamental features, and hence the results are comparable. Both datasets contain the same 41 data features, and 5 different classifications; Normal, DoS, Probe, R2L (Remote to Local) and U2R (User to Root). There are around 800,000 samples in KDD cup 99, and 200,000 samples in NSL-KDD, however many papers chose to train and test on only a fraction of this data to decrease testing times. The datasets are split into training and testing datasets, containing slightly different ratios of attacks. The testing dataset also contains some additional methods of attack not included in the training data, used to test adaptability. These slight differences between training and testing data help test neural networks against overfitting (Divekar, A. and Parekh, M. 2018).

Table 1: Classification frequencies on KDD cup 99 dataset

Classification	Frequency of classification in training dataset (%)	Frequency of classification in testing dataset (%)
Normal	19.69	19.48
DoS	79.24	73.9
Probe	0.83	1.34
R2L	0.23	5.2
U2R	0.01	0.07

The data in table 1 shows the distribution of classification types within the KDD cup 99 dataset. This shows an imbalance towards DoS attacks which can cause a bias towards this classification during training, one of the main criticises of this dataset. This is caused by similar principles to overfitting and often results in Normal data being misclassified as DoS, increasing false positive rates. For this reason, many of the methods I will present are trained on a subset of this data with less bias towards DoS attacks.

Table 2: Classification frequencies on NSL-KDD dataset (total 300,000)

Classification	Frequency of classification in training dataset (%)	Frequency of classification in testing dataset (%)
Normal	53.46	43.08
DoS	36.46	33.08
Probe	9.25	10.74
R2L	0.79	12.22
U2R	0.04	0.89

The data in table 2 shows the distribution of classification types in the NSL-KDD dataset. It was designed as an improvement to the KDD cup 99 dataset with less bias towards DoS attacks, and a greater percentage of Normal data. This helps to reduce false positives caused by learned bias towards DoS attacks.

Table 3: Attacks present in datasets

Classification	Attacks in training and testing datasets	Additional attacks in testing dataset
DoS	back, neptune, smurf, teardrop, land, pod	apache2, mailbomb, rocesstable
Probe	satan, portsweep, ipsweep, nmap	mscan, saint
R2L	warezmaster, warezclient, ftpwrite, guesspassword, imap, multihop, phf, spy	sendmail, named, nmpgetattack, snmpguess, xlock, xsnoop, worm
U2R	rootkit, bufferoverflow, oadmodule, perl	httptunnel, ps, sqlattack, xterm

The data in table 3 shows the different attack types present in both KDD cup 99 and NSL-KDD datasets. This shows that the testing data contains additional attacks which the neural networks have not been trained on. This helps to ensure the neural network has not overfitted to the training data, and tests adaptability to new attack types.

While many argue that these datasets are now outdated, contain redundant data, and have issues with ratios between different classifications, they are still suitable for comparisons. The dataset UNSW-NB15 has been proposed as an alternative, but it lacks adoption and the KDD datasets are still industry standard. If these methods were applied to products for actual networks, they would be trained on a combination of data gathered from the actual network it is to be applied on, along with specifically tailored data containing the different attack types most relevant to that network.

2.6 PROPOSED METHODS

Here I will present the five methods I have examined, with details about how they improve interpretability in neural networks, with positives and negatives of each. In section 5 I will examine how well each method matches my requirements, set in section 4.

2.6.1 Self-Organising Maps (SOM)

The first method I will examine is best detailed in the paper ‘Intrusion Detection System Using Self-Organising Maps’ (Alsulaiman, M., Alyahya, A., Alkharboush, R. and Alghafis, N. 2009). This paper demonstrates a NN-IDS system built using self-organising maps, utilising an architecture type called a Kohonen network, where one layer of the network is 2 dimensional rather than one. This idea was developed as a way to better emulate natural brains, allowing differed areas in the network to specialise to different features. A SOM is a fully connected architecture, although some variations can include convolutional layers (Dozono, H., Niina, G. and Araki, S. 2016),

especially when used for pattern recognition. The main way a SOM differs from a feed forward neural network is in the training algorithm used to alter connection weights. Instead of updating the entire network for each iteration of data, a SOM will select a single neuron which is closest to the current input data feature space. When this neuron is updated to closer match the target output it will also alter its neighbouring neurons by a percentage of the amount it is altered by, depending on their distance from the selected neuron. In this way, features of the input data are seen to "compete" for representation on the network. This results in a neural network where different areas of neurons represent different aspects of the data feature space and can be transcribed into a 2D map representing higher dimensions of features, an example of which could be this:

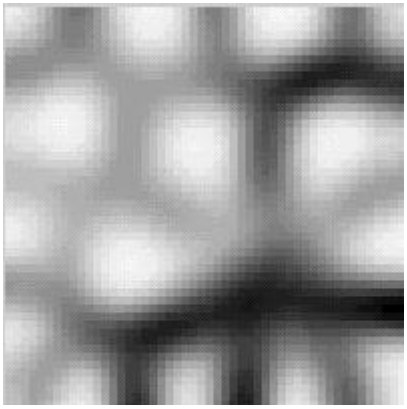


Figure 3: Example output from SOM (Ahn, J. and Syn, S. 2005)

This visual representation of the neural network shows clusters of neurons which have similar neighbours (in white) and neurons which are very different from their neighbours (in black). Each of these clusters represents a different feature or set of features from the data which are considered when making a classification and can aid in understanding how the network breaks down the feature space. Another neural network is then trained to use the two-dimensional representation of the data to make the classification, although this is often integrated directly into the SOM, and the visual representation is extracted from the SOM layer, while the classifier part of the network looks directly at the weights and distances within the SOM layer. This method is used as a way to represent data with higher dimensional feature spaces in a 2D map, but it has the advantage of showing a visual representation of its structure which has been argued as a step towards interpretability (Fortuin, V., Hüser, M., Locatello, F., Strathmann, H. and Rätsch, G. 2019).

2.6.2 Neuro-Fuzzy Classifying

The second method is neuro-fuzzy classifying. This paper (NadjaranToosi, A. (2007) shows how a fuzzy rule system can be applied to intrusion detection to aid in reduction of false positives, however this method of structuring a neural network has potential to aid interpretability (Fernandez, A., Herrera, F., Cordon, O., Jose del Jesus, M. and Marcelloni, F. 2019). Neuro-fuzzy clustering is a method for forcing a neural network to classify features of a dataset using natural language if-

then statements. For example, when examining an image to determine if it contains a cat, it may generate if-then statements like “if these pixels contain this colour” then alter the classification in some way. If the statement is satisfied, then the probability increases that the image contains a cat. The neural network is tasked with deciding what features of the input data satisfy the statements, and to what degree. In this method a human must give the possible classifications beforehand. This is considered a fuzzy logic system as the neural network generates a “score” for how well the input data satisfies each statement, rather than outputting a binary ‘yes’ or ‘no’ result, emulating the way a human brain uses uncertain probabilistic values when making a decision. This method has the advantage of using natural language rules when making classifications, meaning a human is able to read and change the rules, improving interpretability and allowing improvements to be made.

2.6.3 Neuro-Fuzzy Clustering

The next method is similar to neuro-fuzzy classifying, but instead uses fuzzy clustering. The principals are the same; however, the network is not given specific classification labels to sort into. Instead it creates its own unlabelled clusters based on patterns it finds in the data. Despite the end results being similar, the subtle difference means that this method is trained using unsupervised learning, where Neuro-Fuzzy Classification uses supervised training. These different approaches mean the two methods will find different patterns and will result in a completely different network structure. The paper ‘Intrusion Detection using Fuzzy Clustering and Artificial Neural Network’ (Surana, S. 2014) shows how this can be used in intrusion detection, but the principals which make it useful for interpretability are the same as fuzzy classifying. I have included it as a separate solution due to the argument that not defining the classifications before-hand gives the network greater freedom to learn patterns, improving its depth of knowledge and understanding of context, which can aid in generalisation of the network. This can be very useful for explainability if this method was progressed further.

2.6.4 Adversarial Pair of Neural Networks

The fourth method uses an adversarial pair of networks. As shown in the paper ‘An Adversarial Approach for Explainable AI in Intrusion Detection Systems’ (Marino, D., Wickramasinghe, C. and Manic, M. 2018) a second network can be trained alongside a NN-IDS which is tasked with explaining the classifications of the first network (Tomsett, R. 2018). This method has been used previously for penetration testing NN-IDSs and involves training the second network to attempt to figure out the smallest degree of change needed to the least number of input features, to alter the results of the classification. Previously, this was then used to train the IDS to improve its accuracy and prevent attackers doing the same, however this work used the adversary to gain an understanding of what the NN-IDS considers when making the classification. The paper found that by having the adversarial network focus on misclassified data they could generate natural

language statements for what went wrong and show the reasons for the mistake. As an example, the paper shows the generated reasons the NN-IDS misclassified some normal data as a DOS attack:

“Normal samples were mis-classified as DOS because:

- high number of connections to the same host (count) and to the same destination address (dst host count)
- low connection duration (duration)
- low number of operations performed as root in the connection (num root)
- low percentage of samples have successfully logged in (logged in, is guest login)
- high percentage of connections originated from the same source port (dst host same src port rate)
- low percentage of connections directed to different services (diff srv rate)
- low number of connections were directed to the same destination port (dst host srv count)”

Examining these reasons shows why the data was misclassified and gives an understanding of how we can alter the NN-IDS to better classify this data in the future. This technique can be used to both improve the accuracy of the network, and gain a better understanding of classifications, and can be used to help justify any decisions made based on detection.

2.6.5 Layer-Wise Relevance Propagation

The final method I have examined is Layer-Wise Relevance Propagation. It is a technique which has been developed to help generate values showing the relevance of different input features when making a classification. As shown in this paper ‘Improving User Trust on Deep Neural Networks Based Intrusion Detection Systems’ (Amarasinghe, K. and Manic, M. 2018) this method can be used to great success in IDS systems to help improve interpretability. The result is achieved by manually examining the weights in the network through back propagation and developing a ‘heat map’ showing which nodes contributed most to the classification, continuing backwards until you reach the input nodes. There are algorithms to automate the process for larger networks, but they have issues with progressively larger contribution values scaling indefinitely. The advantage of this method is that it can be applied to any neural network architecture to generate basic explanations as to which features were important to the classification, and to what degree.

	Positives	Negatives
Self-Organising Maps	SOM's have been shown to be very effective for use in IDSs due to their ability to easily represent higher dimensions of features in two-dimensional space, making for easier pattern recognition. They have been shown to produce consistently accurate results and have the potential to out-perform standard feed-forward networks in this application. They produce a visual representation of the data, making for easy viewing by a human. This can aid with understanding the results of classifications by examining the types of input features which are considered categorically similar by the network. They can be used in conjunction with other interpretability methods to aid human comprehension of results.	SOM's can be slow to train or evolve to new data features, and it is often faster to completely retrain the network when changes to input features are made. While the visual representation is useful for pattern recognition and improving a classifiers understanding of the data features, they do not present an explanation for how a classification is made without being paired with another method.
Neuro-fuzzy classification	Reading the if-then statements can give a good interpretation of how the network understands and classifies the input data features. The if-then statements can be manually changed by a human if the network misinterprets a data feature allowing us to improve incorrect classifications and overall accuracy. Being able to insert expert knowledge into the system through manipulation of the if-then statements can improve training times of the network while maintaining stability. Fuzzy logic systems excel at categorising data with high dimensionality due to their variable probabilistic scoring of data against the if-then statements, meaning data features which do not clearly show bias towards one classification or another have less chance to affect the result and cause a false positive.	While this method does aid with interpretability by giving a good description of how the input data features effects the classification, it does not aid with explainability, as it is unable to generate reasons for why these features have the effect that they do. This method can take longer to train than other neural networks, especially on data with high dimensionality, and the rule systems can become very complex. Additional manual work is currently needed to understand the classifications by breaking down the if-then statements.
Neuro-fuzzy clustering	Same in most respects to Neuro-Fuzzy Classifying, however using undetermined labels for clusters allows the network to find patterns which would otherwise be lost due to the constraints, which can improve accuracy if the network is trained well.	Same in most respects to Neuro-Fuzzy Classifying, however using undetermined labels for clusters can make training the network unreliable, as it may find patterns which are not relevant to the classifications

	Allowing the network to determine its own classifications can allow it to better learn the context of the data.	we require, and these patterns need to be unlearned to improve accuracy.
An Adversarial Approach for Explainable AI	This method gives easy to read, natural language responses for classifications, meaning a description for why detection was made could accompany the detection in the logs/alerts. This method gives the best representation of how input features are considered, with in depth descriptions for each detection, where other methods only show overall contribution to each detection type. This method can be applied to any other neural network type, meaning the best performing network can be selected, and have an adversarial network paired with it. No additional manual work is needed to extrapolate the information.	Training a second network means trusting a second network which could have additional ethical complications. The second network would have to be fully re-trained if small changes were made to the main IDS network
Layer-wise relevance propagation	This method can be applied to any type of neural network, meaning it can make use of the current leading performance techniques and architectures. It is robust and reliable, giving a clear picture of which nodes in the network, including the input nodes, has the greatest impact on classifications. It is the only method which examines all nodes in the network rather than just input nodes, or a single layer as in the SOM.	This method only generates basic information about the networks structure, and does not give explanations for classifications, or how different values for input features can affect the results. Manually back-propagating takes a lot of time, and the algorithms to automate the process are still ineffective.

2.7 Results of Testing against KDD Datasets

Values in this table are based on preliminary testing and proof of concept models, and were not conducted by me, but by the original owners of the papers proposing each method. Future works will incorporate the described methods using more robust systems and a range of different configurations of the proposed architectures to find optimal solutions. These detection rates and false positive rates are designed to show that these methods can be at least as effective as standard feed-forward neural networks for intrusion detection, with the added benefits of their interpretability qualifier.

Table 4: The results for preliminary testing of methods against KDD dataset

Solution	Detection rates	False Positive rates	Interpretability qualifier
Self-Organising Maps	99%	1.25%	Visual representation of features contributing to classifications
Neuro-fuzzy classification	98.04%	1.88%	A collection of if-then statements showing which input features contribute to which output classifications
Neuro-fuzzy clustering	97%	2.01%	A collection of if-then statements showing which input features contribute to which cluster
An Adversarial Approach for Explainable AI	N/A	N/A	A collection of natural language statements detailing why a classification was made
Layer-wise relevance propagation	N/A	N/A	A heat map type output showing the relevance of each input feature as it propagates through the network

The data in table 4 shows that the three methods tested against the KDD datasets were able to achieve results of >97% accuracy with false positive rates <3%. This shows their ability to match current solutions with further testing and development and shows that using these methods to improve interpretability will not impede the effectiveness of the intrusion detection system.

2.8 SUMMARY

Section 2 has provided a summary of knowledge relating to both IDSs and Neural Networks and has presented the five different solutions which propose to improve interpretability and explainability in NN-IDSs.

3 METHODOLOGY

In this section I will describe the research methodologies I have used to examine the different solutions to the problem. It will show why I chose the method and how it was suitable to my work. I will also describe some of the changes which occurred to my project as I approached completion.

3.1 MoSCoW

During my work I have used the MoSCoW analysis technique. MoSCoW is a prioritisation methodology used to describe the features of a product or solution to a problem. It allows different features to be described based on how important they are to the success of a product, making it clear what needs to be achieved. The prioritisations are split into four sections, Must, Should, Could and Would (Sometimes called Won't). Requirements in the Must category are essential to the completion of the work, and a single requirement from this category missing shows an incomplete or inappropriate product or solution. The Should prioritisation contains the features which are important, but not critical. In most cases all Should requirements will be fulfilled, and a single one missing can be detrimental, but will not invalidate the end result. The Could section is for features which are desirable, but don't make a big difference to the final product. These features are normally included if there is remaining time after the other prioritisations are completed, and an ideal product would have all of these features, however the product is considered complete without them. Items which fall into the Would category are features which are hoped to be in future works. They are normally features which are either too complex, time consuming or undecided on. This section is often used to show future potential of a product, especially if it an early version or using untested technologies or methods.

The MoSCoW methodology is well used in academia and is a trusted method for analysing solutions and products. It is an agile development method, meaning it can show current progress of a project in stages, can be updated during the project and often changes as the project continues. It is a useful tool for looking at solutions to complex problems, especially if the solutions are in early development, as it gives a good estimation as to how complete the solution is and how well it matches the needs of the problem.

I chose this methodology due to the incomplete nature of the field, meaning the solutions to the problem are not yet fully developed. It gives a good estimation of how close each method is to meeting all the criteria of solving the black-box issue. It is useful for comparing solutions as it can quickly show how many points from each prioritisation where fulfilled, putting a numeric score to an otherwise opinion-based comparison.

3.2 Data Types

The research in the paper was carried out using a combination of qualitative and quantitative and research methods. The data gathered on the accuracy of the solutions was quantitative as it was a direct comparison of numeric values representing the scores achieved when testing the solutions on known datasets. Quantitative data is easier to compare without bias, as it shows exact values, however quantitative data often fails to fully summarise the positives and negatives of a solution, so qualitative research can give better representation of features outside the statistics. Because of this, I chose to use a combination of both research types to give a well-rounded, holistic representation of the solutions. One issue with using qualitative research is that it is subject to bias and misrepresentation. This is one reason I chose to use the MoSCoW methodology for my qualitative data, as it gives the same requirements to each method, and gives a clear result as to which methods satisfied more of the points.

Due to this research area being incomplete and largely undeveloped, my sample size is not optimal. For a project like this I would expect to compare around ten different solutions, each with vastly different approaches to the problem. Unfortunately, the only other solutions I was able to find were either too similar to the methods proposed or were not developed far enough to make a fair comparison.

3.3 Project Evolution

During the course of this project, I have altered my goals and objectives to better match the criteria and align better with the expectations of a degree dissertation. My initial plan involved the creation of a neural network-based intrusion detection system of my own design; however, this quickly became unfeasible. This was mainly due to the fact that many such systems exist already, and my knowledge of programming neural networks was not sufficient to improve upon the current leading designs. I experimented with some initial ideas using TensorFlow.js and a few other tools for building neural networks which can utilise GPU acceleration, but the knowledge required for creating an original and successful design was beyond my current abilities.

My next proposal was to conduct a holistic assessment of the current best proposed NN-IDS techniques, looking for systems with the highest detection rates and lowest false positives. This was more realistic as most methods gave in-depth descriptions of their architecture and training methods, which might allow me to recreate and test them. However, after researching the idea it became clear that this has been done many times previously, and I was able to find a number of papers covering this in great detail.

I finally arrived at my chosen topic while browsing news about DeepMinds latest works and came across an article claiming they had ‘Solved’ the black-box problem. At the time I was not well acquainted with the idea of explainability, however the article intrigued me, and I quickly decided to investigate if any research had been done to apply these principals to IDSs. This proved to be a

challenge since only a few papers where available covering the topic, however, I was fortunate to be searching at the time a number of new papers were published, giving me enough material to compare some techniques and gain an appreciation for how the black-box problem related to IDSs and some of the specific challenges in this field.

3.4 SUMMARY

In chapter 3 I have shown why I chose to use the MoSCoW analysis technique, how and why I chose to use both qualitative and quantitate data types in my research and given a brief description of how my current project evolved from my initial ideas.

4 REQUIREMENTS AND ANALYSIS

In this section I will cover the requirements that need to be satisfied to consider my project to be successful, as well as how I will examine each potential solution. It will show what features a solution should have, with a MoSCoW analysis to give an accurate representation for the success of the methods I examine.

4.1 PROJECT SUCCESS CRITERIA

For my project to be successful, I need to examine a range of solutions to the black box issue applied to network intrusion detection, evaluate each solution for both performance and improvements to interpretability and explainability, and show which method or combination of methods can be considered most appropriate based on my requirements. Success for this project does not rely on finding a perfect solution but is to show the current 'state of the art', which solutions have the most potential and recommendations for further works which could build on these solutions.

4.2 IDEAL SOLUTION

The ideal solution for this problem is a neural network technique that gives as good as or improved accuracy for detection rates as current techniques, while also qualifying as Interpretable or Explainable.

The technique will be considered interpretable when it can produce comprehensive, natural language or visual descriptions of its classifications, with details of which input features were considered. These descriptions must be able to aid a human in predicting the output classification based on input values, which gives a better understanding of the cause effect relationship between data and result.

It will be considered explainable when it can describe why it considered those features, how it developed that understanding and can defend its classifications with reason and logic. This will demonstrate an ability to show the networks deeper understanding of the data, how its rational compares to a human and gives us information about how it might react to edge cases or unusual inputs, aiding in the development of a more robust system.

4.3 MOSCOW ASSESSMENT

For a solution to be considered a success there are a number of criteria it can be checked against. The first MoSCoW assessment will show the criteria for an interpretable network, and the second will show additional criteria for a solution to be considered explainable. For a solution to be considered explainable it should first match all criteria from the Must and Should requirements of interpretability.

4.3.1 Interpretability Requirements

4.3.1.1 Must

- The solution must be as accurate as current solutions or show the ability to become as accurate with further development, with a benchmark of 97%
- It must have a false positive rate as low as current solutions or show the ability to reduce the false positive rate with further development, achieving below 3%
- It must have the ability to show which input data features are considered when making a classification.
- It must be able to scale to a network of any size or type.
- It must produce a result which aids in a humans ability to predict the classification based on the input values

4.3.1.2 Should

- It should be able to give a rating for how important each input data feature was when making a classification.
- It should be able to give natural language outputs (or equivalent such as visual representation) when describing features of the data.
- It should be able to give descriptions of how changes to the input data would affect the classification, which can aid in determining how to solve the issue.
- It should be able to give examples for input values which would result in a specific classification

4.3.1.3 Could

- It could have the ability to produce a description or model of ranges of input values which would result in different classifications
- It could have an intuitive user interface allowing for easy monitoring of the system and be able to display detections in a comprehensive manor which does not require security expertise to action upon.
- It could have the ability to be questioned by a user and generate natural language responses to gain further information about the classification.

4.3.1.4 Would

- It would be able to generate potential solutions to detections and action upon those solutions automatically, making it an intrusion prevention system.

4.3.2 Explainability Requirements

4.3.2.1 Must

- It must be able to give explanations for why an input feature changes the classification in the way it does
- It must be able to explain why some input features are more relevant to classifications than other
- It must be able to give a better understanding of its internal mechanics through visual or descriptive aid

4.3.2.2 Should

- It should show an understanding of the context of the data through insight into flaws with the system

4.3.2.3 Could

- It could be able to show an extraction of the deeper knowledge gained through training as a model or natural language explanation

4.3.2.4 Would

- It would be able to give insight into improvements to the security of the network
- It would be able to give details of how to prevent an attack when detected, or given the ability to prevent the attack itself through preventative measures

4.4 SUMMARY

Chapter 4 has set the requirements for any solutions I examine, giving descriptions for features which need to be included for success. It has outlined the methods I will use to asses each proposed method, allowing me to give numeric scores to show the degree to which they solve the problem.

5 RESEARCH FINDINGS AND EVALUATION

In this section I will compare the five proposed solutions to my MoSCoW assessment from section 4 to determine to what degree they satisfy the requirements I have set out.
Since none of the proposed methods are able to meet all the Must and Should requirements for interpretability and none of the explainable requirements have been satisfied I have not included the explainable requirements. This is a reflection of the complexity of the topic of XAI, and the fact that progress in this field is slow compared to the development of AI as a whole.
I decided to combine Neuro-Fuzzy Classifying and Neuro-Fuzzy Clustering due to their similarity, and that they satisfy the same requirements.

Table 5: A comparison of each method against the requirements

	Must	**Should**	**Could**	**Would**
Self-organising maps	SOM's satisfy three of the five **Must** requirements, as it has been shown to outperform standard feed forward networks in both accuracy and false positive rates and can scale to any network size if trained correctly, however it does not make it clear which input features are used in a classification unless paired with another of the examined methods. It also provides no aid to a human attempting to predict the results of the classification.	This method only satisfies one of the four **Should** requirements, as it does produce a visual representation of the data, however it is unable to give a rating for the importance of different features during a classification, nor can it describe how changes to the input data would affect classifications or give examples of which inputs would lead to a specific classification.	Currently none of the **Could** requirements are fulfilled, as a SOM is unable to generate responses to questions from a human or give a range of input values which would cause a specific classification, however it could be integrated into a GUI which would aid simplicity.	This method does not satisfy the **Would** requirement, however it could aid a future method in understanding the features of the input data, improving its utility and accuracy as an intrusion prevention system.

Neuro-Fuzzy Clustering / Classification	Neuro-Fuzzy Classifying and Neuro-Fuzzy Clustering meet four of the five **Must** requirements; it has been shown to be as accurate as standard neural network IDSs even in early testing and shows improvement to false positive rates; it can scale to a network of any size, although it can take longer to train on large networks; it gives a clear natural language representation of which input features are considered for each classification; however, while it can be used to better predict the output this information is difficult to extrapolate and requires manual work to evaluat if the if-then statements are satisfied.	This method satisfies three of the four **Should** requirements as it gives a numerical value when scoring the contribution of different input features. The natural language if-then statements make it easy to understand the reasons for the classification and also make it clear how a classification would change if the input data was altered, though it would be difficult to manually work out for input data with greater dimensionality. However, it is unable to provide examples of inputs which would give specific classifications.	Currently none of the **Could** requirements are satisfied. It is unable to generate responses to questions from a human. Information about detections could be extrapolated from the if-then statements and displayed on a GUI, but this would require further work. It would be possible to work out ranges of results which would cause specific classifications, however this would require a lot of additional work.	This method currently has no preventative systems, meaning it does not meet the **Would** requirement, however information extrapolated from the if-then statements would make deciding on preventative measures easier for an IPS.

Adversarial Pair of Networks	This method meets all five of the **Must** requirements. As it can be applied to any other neural network IDS, it is as accurate as the current best system with as low false of a positive rate and can be used to gain knowledge on how to improve those systems. It scales to a larger network as well as the system it is applied to. It gives clear natural language statements showing exactly which input features contributed to a classification and why. It also gives a clear picture of how inputs correlate to outputs, making it easier to predict the output classification.	This method meets two of the four **Should** requirements. It gives the best explanations for classifications with in-depth descriptions of the input data and shows how the classification would change if the input data was altered. It does not currently show the importance of each input feature for each classification type however this information could be extrapolated manually, or automatically in future works. This method could also be adapted to allow the output of example input data to create a specific output, but this is not currently implemented.	Currently none of the **Could** requirements are satisfied. This method could be adapted to allow for questioning the network with simple queries about detection, by having it manipulate the flagged input data and seeing the result, but this is not currently implemented. The natural language statements would make a GUI far more understandable, with a clear description of each detection, allowing an expert to quickly decide how to react, however this method is still in early testing and is not currently implemented into a finished product. The adversarial network could be altered to test different possibilities of input data to determine a model of how different ranges of inputs can lead to specific outputs, but this is not yet implemented.	This method currently has no preventative systems, meaning it does not meet the **Would** requirement, however information extrapolated from the statements would make deciding on preventative measures easier for an IPS.

Layer-Wise Relevance Propagation	This method meets three of the five **Must** requirements. As it can be applied to any other neural network IDS it is as accurate as the current best system with as low false positive. It gives a clear picture of which input features are important for a classification, with the added advantage of showing hidden layer nodes contributions as well. It partially aids in the ability to predict outputs, but this would require a lot of work tracing how values would change as they propagate through the network. While it can scale to a network of any size and feature dimensionality, it will slow considerably with current algorithms.	This method satisfies one of the four **Should** requirements. It clearly shows the degree of contribution from each input feature; however, it is unable to give any explanations for how these input features are relevant or how changing the results might change the classification. While it can give a rough idea as to which input values could result in a specific classification, it is not detailed or accurate enough to be of much use.	This method meets none of the **Could** requirements. It is unable to generate any explanations for input features or reasons for desertions made by the network. It provides little use for aiding a GUI and can only provide a general idea as to how different inputs would propagate through the network, and not any specific values or ranges.	Layer-wise relevance propagation does not meet the **Would** requirement and provides little aid for a future IPS.

5.1 RESULTS

Table 6: The number of requirements from each prioritisation of MoSCoW assessment satisfied by each solution

	Number of Requirements Satisfied			
Method	**Must**	**Should**	**Could**	**Would**
Self-Organising maps	3	1	0	0
Neuro-Fuzzy Classification	4	3	0	0
Neuro-Fuzzy Clustering	4	3	0	0
Adversarial Pair of Networks	5	2	0	0
Layer-Wise Relevance Propagation	3	1	0	0

The data in table 6 gives a good representation of how each solution compares when assessed against my requirements for interpretability. It shows that while each method has some positives, the methods most accurately matching my criteria is the method utilising an adversarial network trained to give explanations for the first's classifications.

5.2 ANALYSIS

This research has shown that each of these solutions presents valid methods for intrusion detection with accuracy comparable to currently used systems, and while there is not yet a good solution for explainability in this field, some aspects of these methods have made progress towards interpretability. This shows the black-box problem to be unsolved, and still requires significant work in future to achieve an acceptable solution.

The best method I have analysed for aiding interpretability in IDSs is using an adversarial network. The simple natural language statements produced based on why each individual data was classified the way it was is a great improvement over any method showing only which input features contributed to different classification types and provides clear insight into the cause effect relationship between input and output. The statements would make it clear if an attack was a genuine threat or false positive to any experienced user and give a good idea how the attack could be countered.

The method utilising layer-wise relevance propagation had the advantage of showing the internal structure of the network, however it fell short in its ability to give explanations for how the data was relevant to the classification. It was only able to produce a model for how data propagated through

the network for a specific classification type, rather than showing this for each data. It required additional manual work to achieve its results, although automated solutions are possible.
The methods Neuro-Fuzzy Classification and Clustering were both able to produce information which aided the prediction of results by a human; however, this information had to be manually extrapolated from the if-then statements to be useful. This information was almost as good as the statements from an adversarial network, however it lacked the ability to show this for each data and could only show what inputs were contributing to each classification type.
Finally, the visual representation of the input data generated by the self-organising maps gave a useful insight into the patterns and structure of the data, with clear demonstration of how input features were related. While this is useful for a classifier and could potentially make a network more accurate, it does not give a lot of information aiding interpretability. It is useful for understanding the data and can help a human to appreciate how a neural network might view input features, however it does not contribute to solving the black-box issue.

From this work, I conclude that the best method for improving interpretability in a neural network-based network intrusion detection system is to use a combination of three of the proposed methods I have shown. Including a self-organising map or kohonen layer to reduce the dimensionality and complexity of the input data can help with accuracy and pattern recognition as well as be a useful aid for a human to extrapolate information from. Training a second network as an adversary designed to give explanations for each classification based on how the network reacts to changes in the input data generates natural language statements which greatly aid a human's ability to predict the results and shows the relationship between input output correlation and is the best single method currently for improving interpretability. Additionally, using layer-wise relevance propagation to analyse the internal structure of the network can give even greater understanding to how the network classifies data, and helps to alleviate some of the unknown factors of the structure.

5.3 IMPROVEMENTS

There are several ways this solution could potentially be improved without significant changes. The adversarial network could be altered to show additional information such as; producing a model of ranges showing inputs which lead to specific classifications, showing which features are irrelevant to all classifications and giving a score for how confident it is for each classification to aid a human in quickly deciding the likelihood of detection being a false positive. If a better algorithm for automating layer-wise relevance propagation was developed, it could be used to give a heat-map style visual representation for every data input, helping show how data propagated through the network during detection, and could help distinguish any unusual classifications.

5.4 SUMMARY

Section 5 has presented the 5 proposed methods when compared to the requirements from section 4. This has given numeric scores for how well they solve the problem and allows me to compare solutions without bias. It has then analysed the results and given recommendations as to the most appropriate solution or solutions for future work to build on.

6 CONCLUSIONS

6.1 SUMMARY

In this paper I have shown several methods which strive for a greater level of interpretability and explainability in the field of neural network-driven network intrusion detection systems. While none of the proposed methods fully encompass the requirements I set, some were successful in part. From this work I have concluded that an adversarial approach is most suitable for generating reliable explanations for classifications made by an NN-IDS. The ability to be applied to any neural network type is advantageous and allows it to stay on par with the current best systems, and its natural language statements showing exactly which input features contributed to each classification were easily understandable and would give useful insight to a human when a potential detection is made. Based on my definitions I believe this method qualifies as interpretable but falls short on being explainable. The adversary can consistently give insights into which input features led to specific classifications but is unable to explain the logic and reasoning behind them.

From this I conclude that the best solution for creating an interpretable intrusion detection system utilising neural networks is to use a combination of an adversarial network trained to test the reasons for each detection, include a self-organising map or kohonen layer to simplify input data features into a 2 dimensional space, aiding both pattern recognition and accuracy, and giving a human the ability to view features of the data in a better form, and to analyse the network with layer-wise relevance propagation to give a better understanding of the internal structure of the network.

In summary, this work has shown that a great deal more research and development is needed in this field before the black-box issue can be solved, and until it is, we will have to be sceptical of any results produced by a neural network. This does not diminish the usefulness of the technology; however, it does hinder its progress and adoption into modern society and can add risk to using it in intrusion detection, especially applied to critical infrastructure. Without a greater level of understanding as to the deeper knowledge gained by the network, we will be unable to predict how it will react to all situations. If an IDS built upon these technologies was to receive data in a form it has not trained with, or been tested against, it could react in a way which causes harm to the system, either by allowing an attack onto the system or by recommending defensive measures for normal data.

6.2 EVALUATION

Through this work I have successfully shown several methods which could contribute to improving interpretability in neural network-based IDSs, comparing their features and accuracy. I have shown how they improve upon previous solutions which give no indication of how they arrive at a classification and given recommendations as to how future works might build a fully interpretable

NN-IDS. I have explored the limitations of these solutions, showing that they do not yet encompass my definitions for explainability, giving reasons why the field needs further development. I believe I have met each of my objectives, and this work could be useful for future works in furthering the development of NN-IDS systems. However, to be able to call this work compete I would prefer to be able to compare a greater number of solutions, as many as ten if possible, to give a better representation of the possibilities. While I was unable to find as many solutions as I would like, new papers are being published regularly on the topic, and new methods for achieving XAI in IDSs are sure to be developed in the near future.

6.3 Future Work

To improve my work in the future I would like to examine a greater range of solutions, as the current range of methods is limited. As well as this, I would like to attempt to implement some of these solutions myself, building on them and showing their potential through primary research. This would give me a greater understanding of the current 'state of the art' in this field and could give insight into potential improvements. This would also allow me to test these solutions on a level playing field, with the same data sets for each, helping to reduce bias, especially if I were to use a more up-to-date dataset with fewer imbalances, such as UNSW-NB15.

In future research I recommend expanding on the range of information provided by the adversarial network. Adversarial networks are already used as an industry standard for penetration testing of neural network-based intrusion detection systems as they can show methods of altering the input data by small amounts to change the classification type normal, allowing an attack to bypass its detections. By the same principals, this method could be used to show network staff any vulnerability in its current structure and could even automatically update itself to be less susceptible to attacks from an adversarial network. The adversary could also be trained to give a score for how confident the system is in its detections, making it easier to determine the likelihood of a false positive. There are several algorithms proposed to automate the process of layer-wise relevance propagation, and these could allow the heat-map produced to update in real time to each detection. This can help a user to determine if any unusual patterns are spotted in the data and can help show what caused a false positive in situations where the data shows it should have been a normal classification.

Ideally, future works would be able to create a solution which matches the criteria for explainability. While this is a long way off, there are several proposed ideas that could enable this. Most of these solutions revolve around the improvement of generalisation, or even realising artificial general intelligence. It is not yet clear if complete explainability or AGI is even possible, however most researchers are positive, and with advancements in the hardware for running AI, including tensor processing cores and quantum computing solutions, as well as a number of recent achievements for AI in medical, business, gaming and product development, these goals seem closer than ever.

Word count (main body of the report): 11004

Word count (artefact): 3986

Total: 14980

REFERENCES

Adadi, A. and Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI) - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8466590 [Accessed 8 Feb. 2019].

Ahn, J. and Syn, S. (2005). SOM Tutorial. [online] Pitt.edu. Available at: http://www.pitt.edu/~is2470pb/Spring05/FinalProjects/Group1a/tutorial/som.html [Accessed 17 Mar. 2019].

Alsulaiman, M., Alyahya, A., Alkharboush, R. and Alghafis, N. (2009). Intrusion Detection System Using Self-Organizing Maps. [online] Ieeexplore-ieee-org.libezproxy.bournemouth.ac.uk. Available at: https://ieeexplore.ieee.org/document/5319303 [Accessed 18 Mar. 2019].

Amarasinghe, K. and Manic, M. (2018). Improving User Trust on Deep Neural Networks Based Intrusion Detection Systems. [online] researchgate.net. Available at: https://www.researchgate.net/publication/328842396_Improving_User_Trust_on_Deep_Neural_Networks_Based_Intrusion_Detection_Systems [Accessed 1 Feb. 2019].

Amarasinghe, K., Kenney, K. and Manic, M. (2018). Toward Explainable Deep Neural Network Based Anomaly Detection. [online] researchgate.net. Available at: https://www.researchgate.net/publication/326380809_Toward_Explainable_Deep_Neural_Network_Based_Anomaly_Detection [Accessed 19 Mar. 2019].

Bonanno, D. (2017). An approach to explainable deep learning using fuzzy inference. [online] researchgate.net. Available at: https://www.researchgate.net/publication/316708469_An_approach_to_explainable_deep_learning_using_fuzzy_inference [Accessed 3 Feb. 2019].

Choudhury, A. (2019). Explainability V/s Interpretability In Artificial Intelligence. [online] Analytics India Magazine. Available at: https://www.analyticsindiamag.com/explainability-vs-interpretability-in-artificial-intelligence-and-machine-learning/ [Accessed 15 Feb. 2019].

Dickerson, J., Juslin, J., Koukousoula, O. and Dickerson, J. (2001). Fuzzy intrusion detection - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/943772 [Accessed 3 Mar. 2019].

Divekar, A. and Parekh, M. (2018). Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives. [online] researchgate.net. Available at: https://www.researchgate.net/publication/329908471_Benchmarking_datasets_for_Anomaly-based_Network_Intrusion_Detection_KDD_CUP_99_alternatives [Accessed 22 Mar. 2019].

Došilović, F., Brčić, M. and Hlupić, N. (2018). Explainable artificial intelligence: A survey - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8400040 [Accessed 19 Apr. 2019].

Dozono, H., Niina, G. and Araki, S. (2016). Convolutional Self Organizing Map - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7881442 [Accessed 17 Mar. 2019].

Fernandez, A., Herrera, F., Cordon, O., Jose del Jesus, M. and Marcelloni, F. (2019). Evolutionary Fuzzy Systems for Explainable Artificial Intelligence: Why, When, What for, and Where to? - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8610271 [Accessed 21 Apr. 2019].

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H. and Rätsch, G. (2019). SOM-VAE: Interpretable Discrete Representation Learning on Time Series. [online] arXiv.org. Available at: https://arxiv.org/abs/1806.02199 [Accessed 10 Mar. 2019].

Francis, R. (2017). False positives still cause threat alert fatigue. [online] CSO Online. Available at: https://www.csoonline.com/article/3191379/false-positives-still-cause-alert-fatigue.html [Accessed 23 Feb. 2019].

Gall, R. (2018). Machine Learning Explainability vs Interpretability: Two concepts that could help restore trust in AI. [online] Kdnuggets.com. Available at: https://www.kdnuggets.com/2018/12/machine-learning-explainability-interpretability-ai.html [Accessed 24 Feb. 2019].

Georgieva, P. (2016). FUZZY RULE-BASED SYSTEMS FOR DECISION-MAKING. [online] researchgate.net. Available at: https://www.researchgate.net/publication/303373064_FUZZY_RULE-BASED_SYSTEMS_FOR_DECISION-MAKING [Accessed 27 Feb. 2019].

Gunning, D. (2016). Explainable Artificial Intelligence. [online] Darpa.mil. Available at: https://www.darpa.mil/program/explainable-artificial-intelligence [Accessed 20 Mar. 2019].

Hagras, H. (2018). Toward Human-Understandable, Explainable AI - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8481251 [Accessed 22 Mar. 2019].

Keneni, B. (2019). Evolving Rule-Based Explainable Artificial Intelligence for Unmanned Aerial Vehicles - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8612916 [Accessed 21 Feb. 2019].

Landress, A. (2016). A hybrid approach to reducing the false positive rate in unsupervised machine learning intrusion detection - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7506773 [Accessed 20 Feb. 2019].

Marino, D., Wickramasinghe, C. and Manic, M. (2018). An Adversarial Approach for Explainable AI inIntrusion Detection Systems. [online] researchgate.net. Available at: https://www.researchgate.net/publication/328615651_An_Adversarial_Approach_for_Explainable_AI_in_Intrusion_Detection_Systems [Accessed 28 May 2019].

NadjaranToosi, A. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. [online] sciencedirect.com. Available at: https://www.sciencedirect.com/science/article/pii/S0140366407001855 [Accessed 17 Mar. 2019].

Narsingyani, D. and Kale, O. (2015). Optimizing false positive in anomaly based intrusion detection using Genetic algorithm - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7375291 [Accessed 22 Mar. 2019].

Othman, M. and Moh Shan Yau, T. (2017). Neuro Fuzzy Classification and Detection Technique for Bioinformatics Problems - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/4148689 [Accessed 26 Feb. 2019].

Pal, D. and Parashar, A. (2014). Improved Genetic Algorithm for Intrusion Detection System - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7065598 [Accessed 9 Feb. 2019].

Surana, S. (2014). Intrusion Detection using Fuzzy Clustering and Artificial Neural Network. [online] Semanticscholar.org. Available at: https://www.semanticscholar.org/paper/Intrusion-Detection-using-Fuzzy-Clustering-and-Surana/f376e987d4191fc2a015ffc37b35e6621b398ced [Accessed 2 Apr. 2019].

Tomsett, R. (2018). Why the Failure? How Adversarial Examples Can Provide Insights for Interpretable Machine Learning - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8455710 [Accessed 26 Mar. 2019].

Turnbull, M. (2018). Darktrace: When looks aren't everything. [online] A blog from the load balancer experts - Loadbalancer.org. Available at: http://www.loadbalancer.org/blog/darktrace-when-looks-arent-everything/ [Accessed 2 Mar. 2019].

VAN VEEN, F. (2016). The Neural Network Zoo - The Asimov Institute. [online] The Asimov Institute. Available at: https://www.asimovinstitute.org/neural-network-zoo/ [Accessed 3 Feb. 2019].

Wang, G., Hao, J., Ma, J. and Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. [online] sciencedirect.com. Available at: https://www.sciencedirect.com/science/article/pii/S0957417410001417/ [Accessed 14 Apr. 2019].

APPENDIX A – ADDITIONAL INFORMATION

6.4 KEY-WORD DEFINITIONS

Genetic Algorithms are a learning technique based around Charles Darwin's natural selection. It involves the creation of a large set of algorithms, often called a population. These algorithms are written in a way which can be represented by a 'genome' and altering that genome will change some internal part of the algorithm. These algorithms are all tested against the problem and the best performers, as defined by their fitness, are allowed to reproduce. During the reproduction, there are many different techniques that can be used to find a new set of 'children'. Crossover involves taking some amount of the genome from two algorithms and combining them. Mutation allows one small part of the genome to change by a random (most of the time) amount. This is repeated many times until a competent algorithm is found. There is a huge variety of different ways to set up genetic algorithms with different population sizes, mutation rates, crossover functions and how the algorithm is encoded into genes. Genetic algorithms can make use of any machine learning method, and can be applied to countless applications, limited only by processing power and time. Almost any problem that could be solved using other machine learning methods could also be solved with genetic algorithms; however, some applications are more suitable than others.

Neuro-Evolution is the combination of neural networks and genetic algorithms. Neural networks can be expressed as by the number of layers, the number of cells per layer, the activation function per layer or cell, and the weights between those cells and this can all be encoded in a genome. The method of encoding to a genome depends on which values are allowed to be altered during training. More simple examples only encode the weights between cells as the architecture of the network including the layers, cells and activation functions is fixed. Methods that also allow the architecture to change or evolve are called Neuro-Evolution of Augmented Topologies or NEAT. The most famous example of neuro-evolution might be DeepMind's set of neural networks called 'AlphaGo', the first AI to defeat the world go (a complex board game) champion, and recently also defeat professional chess and shogi players, 'AlphaFold', which recently won the 13th 'Critical Assessment of Techniques for Protein Structure Prediction' beating all human and computer algorithms in predicting protein folding, which has huge applications for curing disease, and AlphaStar, the first AI to consistently defeat the world's top StarCraft players. Each of these was considered to be a huge breakthrough and was not predicted to happen for another 50-100 years. Each of these AI trained only by competing with other versions of themselves, in virtual environments playing at thousands of times real speed.

The **Loss function** or cost function of a neural network is used to tell the network how accurate its output is. It is calculated using a number of different methods, but the simplest it just the real

answer (label) minus the output. This is only applicable when the true answer is known, meaning a supervised learning problem. Some of the most common methods for calculating the loss are Mean Squared Error (MSE), Mean Absolute Error (MAE) and Kullback Leibler (KL) Divergence.

The **Fitness function** of a neural network is analogous to the loss function, applied to unsupervised learning or reinforcement learning methods. It is a measure numerical value representing how well the neural network performed. The fitness function is generally unique to the application, for example, it could be progress through a game, distance moved through an obstacle course or any other measure of progress. Finding effective fitness functions can be a huge part of any neural networks development and can have a big impact on performance. Leaving the loss function open to interpretation by saying something along the lines of, a higher loss function is achieved by surviving as long as possible in a game, is what lead early neural networks to develop behaviours such as pausing the game, giving themselves an infinite fitness, or exploiting bugs related to memory management. New works are looking to program 'curiosity' into neural networks by making the fitness function a reflection of how much of the world the network has seen.

An **Optimiser** or optimisation function is the operation used to minimise the loss function. There are a number of different optimisers used for different functions, including stochastic gradient descent (SGD) and 'Adam'. These optimisers calculate to what degree the weights should change throughout the network using 'back propagation' to ensure each layer of the network is optimised. There is some degree of random trial and error during optimisation. An optimisation function also takes a 'learning rate' as an input.

The **Learning rate** of a neural network is a set number applied to the optimisation function. It restricts the size of the changes being made to the weights in the network, so that big leaps are not taken in one direction, only to miss the target. This greatly improves the precision of the results but can lead to increased training time. Many neural networks use a method of adaptive learning rate, where the learning rate decreases as the network approaches the target output.

Back propagation is a series of mathematical steps that allow an error based on the loss function to be propagated backwards through the network. This is necessary to allow the whole network to be optimised rather than just the layer closest to the output.

Mutation rate is the chance of a mutation in the genome when using genetic algorithms. Setting the mutation rate too high leads to random children, were as low mutation rate can stagnate any natural selection if all the parent genomes are too similar.

Adversarial Networks are a field of study that involves training two networks in parallel, each with different goals. The most common example of this is a Generative Adversarial Network (GAN), which is often used in image creation. The main network will attempt to create an image and the second network will try to decide if its real or not. This method has been shown to be incredibly effective in a variety of applications and is the reason 'Deep Fakes' have become so convincing.

A **matrix** is a data object containing numbers in a two-dimensional array. There are a number of mathematical operations that can be applied to matrices allowing manipulation of the values contained. Matrices are crucial in neural networks as they are the method for storing the various weights and nodes of the network.

A **Tensor** is similar to a matrix, except it is inclusive of scalar values (single numbers), vectors (one dimensional arrays), matrices and higher dimensional arrays. They can be manipulated with similar mathematics to matrices.

CPU's, GPU's and TPU's are all forms of processing units which specialise in a specific type of math. CPU's or central processing units focus on single thread calculations that are important for running programs and operating systems, GPU's or graphics processing units focus on matrix manipulation which is used for graphics rendering and TPU's or tensor processing units are configured for tensor manipulation, making them suitable for running neural networks. Each of these is capable of the math for neural networks to function, but each is an order of magnitude more efficient than the last.

Overfitting is when a neural network is trained on a small dataset, trained for too long, or the dataset contains to little variety and or noise. This causes the neural network to learn exact patterns in the dataset rather than general rules, which will cause it to struggle with new data. This is one reason datasets are broken up into training data and testing data, so results can be procured from data the network has not trained on.

Generalisation is the degree to which a solution can adapt to different problems and is the first step towards AGI. A more generalised neural network would be of greater use for explainability due to the necessity of understanding the context of the data.

6.5 Additional Information About Different Cell Types

- A **recurrent cell** operates in the same way as a hidden cell, and can use any type of activation function, with the only difference being its output is fed back to itself after n number of iterations. This gives it the ability for understanding context, where the previous

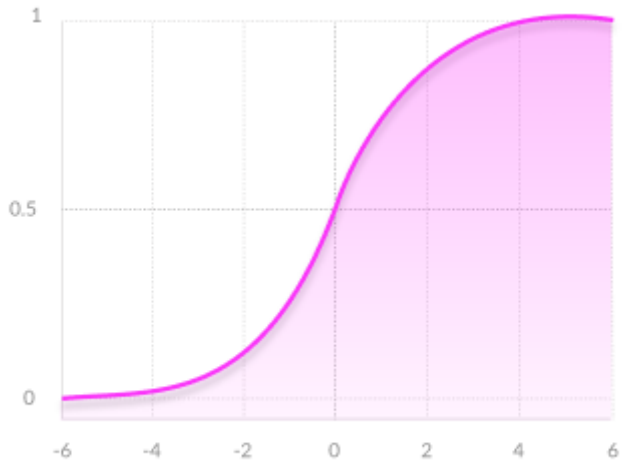
data might affect the current data. For example, text generation requires knowledge of how the previous words in a sentence effect what should be written next.

- **Memory cells** are the first example of greater complexity within cells. They are comprised of a number of elements called gates. Each gate can contain its own activation function and weight and combined they allow the cell to decide whether to forget, remember, pass back or pass forward outputs at any given time. They have a similar function to a recurrent cell, except they pass the output value back to the cell after an arbitrary length of time rather than number of iterations. They are commonly used for video generation.
- **Different memory cells** are just memory cells with slightly altered gates. In **figure x** they are used in a Gated Recurrent Unit, and in this case the different memory cells are memory cells with no output gate, allowing for more efficient processing in some cases, such as speech synthesis.
- **Match Input Output cells** are output cells in which the result should just be an altered form of the input. They are used most commonly in autoencoders.
- **Noisy Input cells** are designed to add a small amount of random variation to the input data. This is to combat a problem with neural networks called 'overfitting'. They are often used where the input dataset is either small or has little variation, or in cases where the data in the real world would be subject to noise but the training data is clean. Chaotic (pseudo-random), thermal, or quantum noise can be used for different situations.
- **Probabilistic Hidden cells** use a radical bias function for their activation function but apply it to the difference between the result and the mean of the cells value.
- **Back-Fed Input cells** provide a point where feedback can be introduced to the network.
- **Kernel cells**, **Convolutional cells** and **Pooling Layers** are features of a convolutional neural network (CNN). Combined they allow a specific part of the network to process only a specific section of the input. This is commonly used when some parts of the input are radically different from others, and the different parts have no effect on each other. This is commonly used for image classification and allows specific parts of the network to focus on specific features within the image.
-
- **Spiking Hidden cells** use a type of activation function which more closely matches the workings of a biological neural network. Networks build with spiking cells tend to run for a specific length of time rather than number of iterations. They are used to imitate biological networks rather than for specific problems.
- A layer is considered '**dense'** or **fully connected** if all inputs connect to all outputs in that layer, and a network is fully connected if it contains only fully connected layers. Some layers such as convolutional layers are inherently not fully connected, while other layer types can be fully connected or not depending on desired effect.

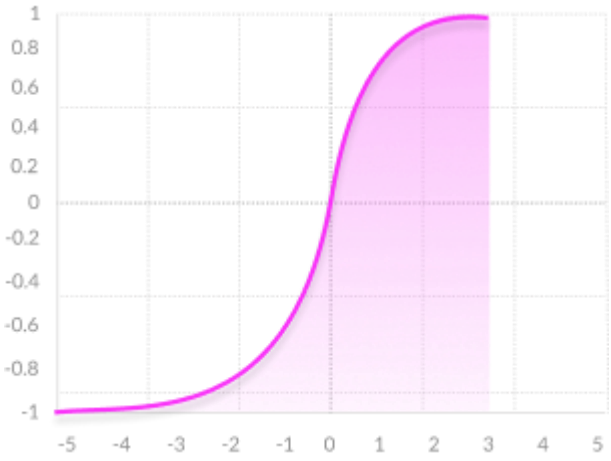
6.6 Activation Functions

A cells activation function is the mathematical operation it applies to the weights inputted to it. There are a number of different types of activation function:

- **Binary step function** - If the input is above a specific value, output one, else output zero.
- **Linear activation function** - Takes input values, multiplied by the weights and gives an output value directly proportional to the input.
- **Non-Linear activation function** – Allow complex mapping of inputs multiplied by weights, to values between a range, giving values closer to a specific part of that range increased weight.
    - **Sigmoid/Logistic** – Maps inputs to values between 1 and 0, with smooth gradient

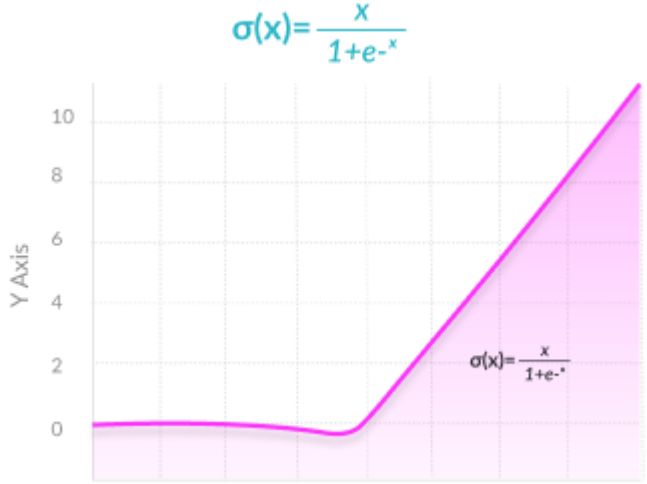


    - **TANH/Hyperbolic tangent** – same as sigmoid but maps from -1,1



    - **SoftMax** – maps inputs multiplied by weights to values using any other activation function, then divides by total output from that layer, forcing all the outputs to sum to 1. Used as the output layer in classification problems so outputs are equal to the percentage chance the network estimates a given classification.

- **SWISH** – a new, largely untested activation function which is very computationally efficient, and produces high accuracy in classification problems.



- There are many other non-linear activation functions being developed and tested but these are the most common or most promising currently.

APPENDIX B – PROJECT PROPOSAL

BU Computing Programmes 2018-2019

Undergraduate Project Proposal Form

Please refer to the **Project Handbook Section 4** when completing this form

Degree Title: BSc (Hons) Forensic Computing and Security	**Student's Name:** Sam Denton
	Supervisor's Name: Neetesh Saxena
	Project Title/Area: A comparison of neural network-based network IDS techniques for achieving interpretability

Section 1: Project Overview

1.1 Problem definition - use <u>one sentence</u> to summarise the problem:

Many network intrusion detection systems have turned to using machine learning technologies like neural networks to power their systems, however, these technologies have the drawback of being 'black-box' in nature, meaning we don't understand the internal mechanics of the system.

1.2 Project description - briefly explain your project:

This project will highlight the need for interpretability and explainability in neural network-based IDS's, showing how their current black-box nature means we are unable to predict how they might react to new situations. This is a big issue as this technology has been applied to a vast number of systems, including critical infrastructure, and a misclassification could lead to attacks going unnoticed if the data is not formatted in the same way the NN-IDS was trained.

Fortunately, the field of explainable artificial intelligence (XAI) exists to attempt to dissect the inner mechanics of neural networks to give a better understanding of how they work, the depth of knowledge they possess and how they will react to edge cases. The field is split into two main sections, interpretability and explainability. In this work I will describe the definitions for both and give requirements for how each can be satisfied.

During this project I will analyse a number of different proposed methods for improving interpretability and explainability, how these methods apply to network intrusion detection and give descriptions for how they function. I will compare each to the requirements I set out and show which is most suitable as a solution to the black-box problem. I will also show potential improvements future research could direct towards and give my own recommendations for creating a NN-IDS matching the criteria for interpretability and explainability.

1.3 Background - please provide brief background information, e.g., client:

This work will be aimed towards aiding future development of NN-IDS's which have a greater focus on interpretability and explainability, with recommendations as to the best methods currently proposed. This research field is still in very early development and so this work will give a holistic overview of current methods, which will aid anyone looking to develop this technology further. The eventual outcome of this research is to create a fully explainable NN-IDS, which will be of great value to any business or critical infrastructure to which it is applied.

BU Computing Programmes 2018-2019

1.4 Aims and objectives – what are the aims and objectives of your project?
AIMS:
To research the 'state of the art' in solutions for the black box issue within Artificial Intelligence driven Network Intrusion Detection Systems, determine their suitability and recommend the most fitting solutions from current proposed designs, with potential improvements.
OBJECTIVES:
-Explore current proposed solutions to the black box problem in neural network-based network intrusion detection systems
-Determine the positives and negatives of each method, pertaining to how they affect detection rates and how well they fit my definitions for interpretability and explainability
-Evaluate solutions to determine best fit to solve the problem, give recommendation for which solution is best and provide insight into potential improvement

Section 2: Artefact

2.1 What is the artefact that you intend to produce?
My artefact will consist of a research paper examining the positives and negatives of a number of different solutions to the black-box problem. It will show how each functions and performs when tested against standard datasets, as well as how each contributes towards developing XAI. It will show the architecture of the proposed method, as well as how this can be applied to a NN-IDS.
It will compare each of the methods to a set of requirements to show how well they satisfy my definitions for interpretable and explainable. It will then summarise my findings and show which method or methods was most acceptable in solving the problem, with details to its successes and shortcoming. It will also give examples as to how the method could be improved in future work to further aid the development of XAI in IDS's.
It will summarise to what degree my final solution solves the black-box issue, and what research is still required.
2.2 How is your artefact actionable (i.e., routes to exploitation in the technology domain)?
My artefact hopes to provide useful information to future projects and will show detailed assessments of the practicality and feasibility of different types of neural networks when applied to intrusion detection systems. It will help direct future research into the most promising avenues of the technology, showing how I believe interpretability and explainability will be achieved in NN-IDS's.
As most developments in artificial intelligence are made open-source, this project is more interested in aiding in the development and advancement of artificial intelligence in the security sector, as well as widespread adoption of the technology, rather than in creating a financially viable product.

BU Computing Programmes 2018-2019

Section 3: Evaluation

3.1 How are you going to evaluate your work?

I will asses the state of my project by comparing it to my aims and objectives and deciding if I believe I have met those criteria. My project does not rely on finding a perfect solution to the problem to be considered complete, rather I will evaluate its completion based on weather it gives a holistic overview of the currently proposed solutions to the problem. It will be required to give a fair and detailed representation of each solution, with a final verdict as to the most appropriate solution, showing how it solves the problem and its shortcomings.

3.2 Why is this project honours worthy?

This product is worthy of being considered Batchelors Degree level due to the in-depth knowledge needed in both the field of intrusion detection and artificial intelligence and will require significant leaning and research on my part. It is aiming to solve a well-documented problem which currently has no solution. If this work is successful, it could contribute to the improvement of all NN-IDS's, and help propagate AI further into the field of cyber security and modern society at large.

3.3 How does this project relate to your degree title outcomes?

Intrusion detection is a big part of the security sector and having the chance to conduct in-depth research into the field will give me significant knowledge into a verity of relevant topics. My project has the potential to provide useful research and insight into how the security sector and artificial intelligence are merging.

3.4 How does your project meet the BCS Undergraduate Project Requirements?

This project meets the requirements as it involves significant knowledge of various FSC subjects, including intrusion detection and prevention, artificial intelligence and machine learning as well as presenting opportunities for problem solving. It aims to contribute to solving a real-world problem and hopes to advance adoption and development of a relatively young technology, artificial intelligence.

3.5 What are the risks in this project and how are you going to manage them?

One risk for this project is the rapid pace of progression in both artificial intelligence technologies and the field of security. In the months it takes to complete this work many new advancements are predicted to occur, and AI in particular has always exceeded expectations in its rapid developments. These could potentially make this work obsolete, as the methods I describe could conceivably be outdated by the time I am complete. I can mitigate this risk by including in my research, time for keeping up to date on the latest developments and either adapting my work or using it as evidence in the projects favour.

One risk that affects many AI projects is that classical computing techniques and algorithms might be considered good enough, at least for now, following the mentality of "If it ain't broke, don't fix it". This seems unlikely since many intrusion detection systems have already adopted machine learning techniques, however advancements in classical computing methods could outpace AI development. This does not greatly affect the progress of my work, as interpretability will be useful for any future projects which choose to use neural networks.

Another risk is that interpretability may not be considered important enough to devote additional research to, and efforts could be diverted into improving the accuracy, speed and generalisation of the neural networks. This is a financial driven risk as business may care more about the IDS being robust and accurate than being able to explain its actions.

BU Computing Programmes 2018-2019

From a personal health standpoint, spending upwards of eight hours a day working on a computer, especially a single project, can lead to back and eye strain, and can cause issues with sleeping. This can be mitigated by using a comfortable, upright seating position, taking regular breaks away from the screen and exercising regularly. I will also be wearing glasses which block harsh blue light from monitors which can aid in reducing eye strain and can help avoid sleep disruption.

Section 4: References

4.1 Please provide references if you have used any.

Adadi, A. and Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI) - IEEE Journals & Magazine. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8466590 [Accessed 8 Feb. 2019].

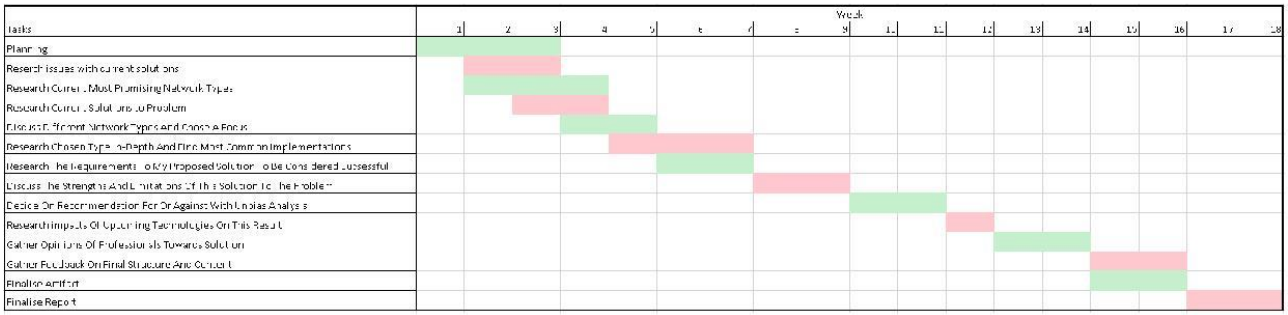
Došilović, F., Brčić, M. and Hlupić, N. (2018). Explainable artificial intelligence: A survey - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8400040 [Accessed 19 Apr. 2019].

Gunning, D. (2016). Explainable Artificial Intelligence. [online] Darpa.mil. Available at: https://www.darpa.mil/program/explainable-artificial-intelligence [Accessed 20 Mar. 2019].

Section 5: Ethics (please delete as appropriate)

5.1 Have you submitted the ethics checklist to your supervisor? **Yes**
5.2 Has the checklist been approved by your supervisor? **Yes**

Section 6: Proposed Plan (please attach your Gantt chart below)



Edited by Dr Nan Jiang and Dr Deniz Cetinkaya based on PH Section 4

APPENDIX C – RESEARCH ETHICS CHECKLIST

Bournemouth University

Research Ethics Checklist

About Your Checklist	
Reference Id	27295
Date Created	28/05/2019 13:54:08
Status	Approved
Date Approved	28/05/2019 14:34:58
Date Submitted	28/05/2019 14:22:13

Researcher Details	
Name	Sam Denton
Faculty	Faculty of Science & Technology
Status	Undergraduate (BA, BSc)
Course	BSc (Hons) Forensic Computing & Security
Have you received external funding to support this research project?	No

Project Details	
Title	A comparison of neural network-based network IDS techniques for achieving interpretability
Start Date of Project	31/01/2019
End Date of Project	31/05/2019
Proposed Start Date of Data Collection	31/01/2019
Supervisor	Neetesh Saxena
Approver	Neetesh Saxena

Summary - no more than 500 words (including detail on background methodology, sample, outcomes, etc.)

In this paper I will show some of the current leading techniques for improving interpretability and explainability, how they can be applied to IDS's, and give detailed explanations as to how they can improve both the field of deep learning and cyber security. I will also put emphasis on ensuring the techniques examined do not impede the accuracy of the system, and as such will include figures produced by testing preliminary models utilising the proposed techniques. The purpose of this work is to aid in the development of future IDS's which have an emphasis on interpretability and explainability. It is intended to show which method or methods show most promise in developing this field, and direct further research towards these techniques.

To fully appreciate the differences between solutions examined in this paper, as well as what I am looking for in an ideal solution, it must first be clear what the definitions of interpretability and explainability are, and the differences between them.

Interpretability refers to the extent to which we can describe the cause effect relationship between the input data and output result. It has also been described as our ability to predict the output of the neural network based on changes to the input.

Explainability expands on this by showing the extent to which we understand the internal mechanics of a neural network and can describe the networks reasoning behind classifications. This does not simply mean showing the layout of nodes and connections but means being able to extrapolate deeper meaning and learned knowledge from the network.

The difference is subtle, but explainability requires a much deeper level of comprehension. For example, an interpretable neural network might be able to tell you which features of a dataset it considers to be important, while an explainable neural network could tell you why those features are important, and answer questions about the process leading to the development of that knowledge.

In an ideal solution this might present itself in a few ways. An interpretable neural network might be able to produce a visual or natural language representation of which input features it considers to be important for a classification. It might be able to describe how changing different input values would change the classification. It could give examples of what input value ranges would result in a specific classification. An explainable neural network should be able to take this a step further and be able to show an ability to answer 'why' questions. Why is this input feature important? Why did changing this input have the effect that it did? Once these questions can be answered, we can begin to assume it is an explainable neural network.

During my work I have used the MoSCoW analysis technique. MoSCoW is a prioritisation methodology used to describe the features of a product or solution to a problem. It allows different features to be described based on how important they are to the success of a product, making it clear what needs to be achieved. The prioritisations are split into four sections, Must, Should, Could and Would (Sometimes called Won't). Requirements in the Must category are essential to the completion of the work, and a single requirement from this category missing shows an incomplete or inappropriate product or solution. The Should prioritisation contains the features which are important, but not critical. In most cases all Should requirements will be fulfilled, and a single one missing can be detrimental, but will not invalidate the end result. The Could section is for features which are desirable, but don't make a big difference to the final product. These features are normally included if there is remaining time after the other prioritisations are completed, and an ideal product would have all of these features, however the product is considered complete without them. Items which fall into the Would category are features which are hoped to be in future works. They are normally features which are either too complex, time consuming or undecided on. This section is often used to show future potential of a product, especially if it an early version or using untested technologies or methods.

The MoSCoW methodology is well used in academia and is a trusted method for analysing solutions and products. It is an agile development method, meaning it can show current progress of a project in stages, can be updated during the project and often changes as the project continues. It is a useful tool for looking at solutions to complex problems, especially if the solutions are in early development, as it gives a good estimation as to how complete the solution is and how well it matches the needs of the problem.

Literature Review

Additional Details	
Will you have access to personal data that allows you to identify individuals which is not already in the public domain?	No
Will you have access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or separate confidentiality agreement)?	No

Storage, Access and Disposal of Personal Data	
Will any data be stored on the BU's Data Repository "BORDaR"?	No

APPENDIX D – ARTIFACT

A comparison of neural network-based network IDS techniques for achieving interpretability

Sam Denton

Science and Technology Department, Bournemouth University

ABSTRACT

Deep learning and neural network techniques have proven to be an effective solution for network intrusion detection, and look set to dominate the field of cyber security. However, these technologies suffer a major drawback, their black box nature. The inability to interpret, explain and understand how neural networks reach classification results hinders our ability to trust and rely on them for important tasks such as intrusion detection, especially when applied to critical infrastructure. Lack of understanding makes it hard to be sure that deep learning and neural network techniques will transition reliably from controlled testing environments to working situations, were the results from their classifications will mean the difference between an attack being detected or allowed into a network, with potentially catastrophic consequences.

In this paper I will examine some of the current 'state of the art' techniques which can improve on interpretability and explainability when applied to network intrusion detection systems, looking at how they affect the systems accuracy and false positive rates, as well as giving detailed explanations as to how they aid our understanding of the systems.

1 INTRODUCTION

1.1 PROBLEM DEFINITION

Since the first neural networks were proposed in 1943, it has been apparent that, while this technology has the potential to solve problems beyond the scope of a human brain and standard computation algorithms, we struggle to explain and interpret how the network reached the solution. This is an issue because many of the problems these networks are tackling could have significant impact on the human population, and to be able to trust the results, we need to understand why they are being made. Being able to understand the results also allows us to correct mistakes or misclassifications and build more powerful and reliable neural networks. If the neural network could explain its classifications and hence detections in comprehensible natural language, we would be able to quickly understand what caused the detection, and if additional action is required. This is invaluable, both to generate greater levels of trust in the system, but also identify false positives, of which IDSs tend to show many. A quote from the paper 'Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)' (Adadi, A. and Berrada, M. 2018) talks about the limitations of using black-box systems, despite their promise:

> 'even with such unprecedented advancements, a key impediment to the use of AI-based systems is that they often lack transparency. Indeed, the black-box nature of these systems allows powerful predictions, but it cannot be directly explained'.

This sums up well the current state of almost all systems utilising machine learning techniques like neural networks and shows the need for greater levels of interpretability in our future systems.

Because of the desire to understand neural networks, the field of explainable artificial intelligence (interpretable AI, XAI) is almost as old as AI itself. Not much progress was made in the field until the last 10 years, but it is becoming apparent that for AI to progress further into modern society, we must learn to trust it, and the first step to trust is understanding. Because of this, there has been a big push for XAI and many techniques have been developed. The Defence Advanced Research Projects Agency (DARPA) has expressed great interest in the field, and in 2016 created a program specifically for aiding and funding the progression of XAI. Quoting their official website:

> "the effectiveness of these systems is limited by the machine's current inability to explain their decisions and actions to human users. Explainable AI—especially explainable machine learning—will be essential to understand, appropriately trust, and effectively manage an emerging generation of artificially intelligent machine partners" (Gunning, D. 2016).

1.2 IMPACT

This problem is especially relevant when neural networks are applied to network intrusion detection systems, and even more so when we wish to apply them to critical infrastructure. These systems are the backbone for many important aspects of modern civilisation such as banking, communication, medical care and many others and as such it is imperative that we understand and trust the systems protecting them (Amarasinghe, K. and Manic, M. 2018) (Amarasinghe, K., Kenney, K. and Manic, M. 2018).

Regardless of how accurate and robust the systems become, there will always be edge cases which have not been tested against. If an attack contained data which the network had never seen before, we have no way of ensuring it would react in a suitable manner, since we don't understand how it evaluates the data. This could have catastrophic affects as an attack on critical infrastructure which is not detected could cause irreparable damage, and potentially be life threatening. Less dangerous, but maybe more likely is the opposite. If normal but unusual data is shown as an attack, time and money could be wasted trying to prevent something which does not exist.

1.3 AIMS:

To research the 'state of the art' in solutions for the black box issue within Artificial Intelligence driven Network Intrusion Detection Systems, determine their suitability and recommend the most fitting solutions from current proposed designs, with potential improvements.

1.4 OBJECTIVES:

1. Explore current proposed solutions to the black box problem in neural network-based network intrusion detection systems
2. Determine the positives and negatives of each method, pertaining to how they affect detection rates and how well they fit my definitions for interpretability and explainability
3. Evaluate solutions to determine best fit to solve the problem, give recommendation for which solution is best and provide insight into potential improvement

1.5 EXPLAINABILITY VS INTERPRETABILITY

To fully appreciate the differences between solutions examined in this paper, as well as what I am looking for in an ideal solution, it must first be clear what the definitions of interpretability and explainability are, and the differences between them (Choudhury, A. 2019) (Došilović, F., Brčić, M. and Hlupić, N. 2018) (Gall, R. 2018).

1.5.1 Interpretability

Interpretability refers to the extent to which we can describe the cause effect relationship between the input data and output result. It has also been described as

our ability to predict the output of the neural network based on changes to the input.

1.5.2 Explainability

Explainability expands on this by showing the extent to which we understand the internal mechanics of a neural network and can describe the networks reasoning behind classifications. This does not simply mean showing the layout of nodes and connections but means being able to extrapolate deeper meaning and learned knowledge from the network.

2 PROPOSED METHODS

Here I will present the five methods I have examined, with details about how they improve interpretability in neural networks.

2.1.1 Self-Organising Maps (SOM)

The first method I will examine is best detailed in the paper 'Intrusion Detection System Using Self-Organising Maps' (Alsulaiman, M., Alyahya, A., Alkharboush, R. and Alghafis, N. 2009)**.** This paper demonstrates a NN-IDS system built using self-organising maps, utilising an architecture type called a Kohonen network, where one layer of the network is 2 dimensional rather than one. This idea was developed as a way to better emulate natural brains, allowing differed areas in the network to specialise to different features. A SOM is a fully connected architecture, although some variations can include convolutional layers (Dozono, H., Niina, G. and Araki, S. 2016), especially when used for pattern recognition. The main way a SOM differs from a feed forward neural network is in the training algorithm used to alter connection weights. Instead of updating the entire network for each iteration of data, a SOM will select a single neuron which is closest to the current input data feature space. When this neuron is updated to closer match the target output it will also alter its neighbouring neurons by a percentage of the amount it is altered by, depending on their distance from the selected neuron. In this way, features of the input data are seen to "compete" for representation on the network. This results in a neural network where different areas of neurons represent different aspects of the data feature space and can be transcribed into a 2D map representing higher dimensions of features, an example of which could be this:

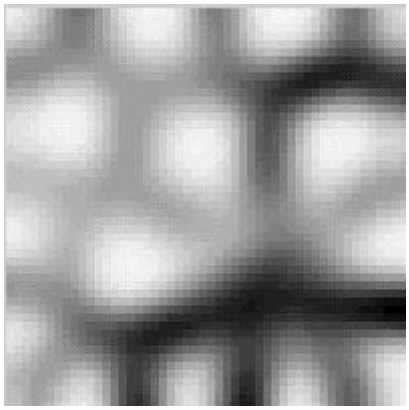


Figure 1: Example output from SOM (Ahn, J. and Syn, S. 2005)

This visual representation of the neural network shows clusters of neurons which have similar neighbours (in white) and neurons which are very different from their neighbours (in black). Each of these clusters represents a different feature or set of features from the data which are considered when making a classification and can aid in understanding how the network breaks down the feature space. Another neural network is then trained to use the two-dimensional representation of the data to make the classification, although this is often integrated directly into the SOM, and the visual representation is extracted from the SOM layer, while the classifier part of the network looks directly at the weights and

distances within the SOM layer. This method was designed as a way to represent data with higher dimensional feature spaces in a 2D map, but it has the advantage of showing a visual representation of its structure which has been argued as a step towards interpretability (Fortuin, V., Hüser, M., Locatello, F., Strathmann, H. and Rätsch, G. 2019).

This method is most useful in aiding interpretability if paired with another method like an adversarial pair or layer-wise relevance propagation which can show exactly which input features are considered during a classification as it makes the results of these methods easier to read by a human.

2.1.2 Neuro-Fuzzy Classifying

The second method is neuro-fuzzy classifying. This paper (NadjaranToosi, A. (2007) shows how a fuzzy rule system can be applied to intrusion detection to aid in reduction of false positives, however this method of structuring a neural network has potential to aid interpretability (Fernandez, A., Herrera, F., Cordon, O., Jose del Jesus, M. and Marcelloni, F. 2019). Neuro-fuzzy clustering is a method for forcing a neural network to classify features of a dataset using natural language if-then statements. For example, when examining an image to determine if it contains a cat, it may generate if-then statements like “if these pixels contains this colour” then alter the classification in some way. If the statement is satisfied, then the probability increases that the image contains a cat. The neural network is tasked with deciding what features of the input data satisfy the statements, and to what degree. In this method a human must give the possible classifications beforehand. This is considered a fuzzy logic system as the neural network generates a “score” for how well the input data satisfies each statement, rather than outputting a binary ‘yes’ or ‘no’ result, emulating the way a human brain uses uncertain probabilistic values when making a decision. This method has the advantage of using natural language rules when making classifications, meaning a human is able to read and change the rules, improving interpretability and allowing improvements to be made.

2.1.3 Neuro-Fuzzy Clustering

The next method is similar to neuro-fuzzy classifying, but instead uses fuzzy clustering. The principals are the same, however the network is not given specific classification labels to sort into. Instead it creates its own unlabelled clusters based on patterns it finds in the data. Despite the end results being similar, the subtle difference means that this method is trained using unsupervised learning, where Neuro-Fuzzy Classification uses supervised training. These different approaches mean the two methods will find different patterns and will result in a completely different network structure. The paper ‘Intrusion Detection using Fuzzy Clustering and Artificial Neural

Network' (Surana, S. 2014) shows how this can be used in intrusion detection, but the principals which make it useful for interpretability are the same as fuzzy classifying. I have included it as a separate solution due to the argument that not defining the classifications before-hand gives the network greater freedom to learn patterns, improving its depth of knowledge and understanding of context, which can aid in generalisation of the network. This can be very useful for explainability if this method was progressed further.

2.1.4 Adversarial Pair of Neural Networks

The fourth method uses an adversarial pair of networks. As shown in the paper 'An Adversarial Approach for Explainable AI in Intrusion Detection Systems' (Marino, D., Wickramasinghe, C. and Manic, M. 2018) a second network can be trained alongside a NN-IDS which is tasked with explaining the classifications of the first network (Tomsett, R. 2018). This method has been used previously for penetration testing NN-IDSs and involves training the second network to attempt to figure out the smallest degree of change needed to the least number of input features, to alter the results of the classification. Previously, this was then used to train the IDS to improve its accuracy and prevent attackers doing the same, however this work used the adversary to gain an understanding of what the NN-IDS considers when making the classification. The paper found that by having the adversarial network focus on misclassified data they could generate natural language statements for what went wrong and show the reasons for the mistake. As an example, the paper shows the generated reasons the NN-IDS misclassified some normal data as a DOS attack:

"Normal samples were mis-classified as DOS because:

- high number of connections to the same host (count) and to the same destination address (dst host count)
- low connection duration (duration)
- low number of operations performed as root in the connection (num root)
- low percentage of samples have successfully logged in (logged in, is guest login)
- high percentage of connections originated from the same source port (dst host same src port rate)
- low percentage of connections directed to different services (diff srv rate)
- low number of connections were directed to the same destination port (dst host srv count)"

Examining these reasons shows why the data was misclassified and gives an understanding of how we can alter the NN-IDS to better classify this data in the future. This technique can be used to both improve the accuracy of the network, and gain a better understanding of classifications, and can be used to help justify any decisions made based on detection.

2.1.5 Layer-Wise Relevance Propagation

The final method I have examined is Layer-Wise Relevance Propagation. It is a

technique which has been developed to help generate values showing the relevance of different input features when making a classification. As shown in this paper 'Improving User Trust on Deep Neural Networks Based Intrusion Detection Systems' (Amarasinghe, K. and Manic, M. 2018) this method can be used to great success in IDS systems to help improve interpretability. The result is achieved by manually examining the weights in the network through back propagation and developing a 'heat map' showing which nodes contributed most to the classification, continuing backwards until you reach the input nodes. There are algorithms to automate the process for larger networks, but they have issues with progressively larger contribution values scaling indefinitely. The advantage of this method is that it can be applied to any neural network architecture to generate basic explanations as to which features were important to the classification, and to what degree.

3 REQUIREMENTS

For a solution to be considered a success there are a number of criteria it can be checked against.

Must

- The solution must be as accurate as current solutions or show the ability to become as accurate with further development, with a benchmark of 97%
- It must have a false positive rate as low as current solutions or show the ability to reduce the false positive rate with further development, achieving below 3%
- It must have the ability to show which input data features are considered when making a classification.
- It must be able to scale to a network of any size or type.
- It must produce a result which aids in a humans ability to predict the classification based on the input values

Should

- It should be able to give a rating for how important each input data feature was when making a classification.
- It should be able to give natural language outputs (or equivalent such as visual representation) when describing features of the data.
- It should be able to give descriptions of how changes to the input data would affect the classification, which can aid in determining how to solve the issue.
- It should be able to give examples for input values which would result in a specific classification

Could

- It could have the ability to produce a description or model of ranges of input values which would result in different classifications
- It could have an intuitive user interface allowing for easy monitoring of the system and be able to display detections in a comprehensive manor which does not require security expertise to action upon.
- It could have the ability to be questioned by a user and generate natural language responses to gain further information about the classification.

Would

- It Would be able to generate potential solutions to detections and action upon those solutions automatically, making it an intrusion prevention system.

4 RESEARCH FINDINGS AND EVALUATION

Table 1: A comparison of each method against the requirements

	Must	**Should**	**Could**	**Would**
Self-organising maps	SOM's satisfy three of the five **Must** requirements, as it has been shown to outperform standard feed forward networks in both accuracy and false positive rates and can scale to any network size if trained correctly, however it does not make it clear which input features are used in a classification unless paired with another of the examined methods. It also provides no aid to a human attempting to predict the results of the classification.	This method only satisfies one of the four **Should** requirements, as it does produce a visual representation of the data, however it is unable to give a rating for the importance of different features during a classification, nor can it describe how changes to the input data would affect classifications or give examples of which inputs would lead to a specific classification.	Currently none of the **Could** requirements are fulfilled, as a SOM is unable to generate responses to questions from a human or give a range of input values which would cause a specific classification, however it could be integrated into a GUI which would aid simplicity.	This method does not satisfy the **Would** requirement, however it could aid a future method in understanding the features of the input data, improving its utility and accuracy as an intrusion prevention system.

Neuro-Fuzzy Clustering / Classification	Neuro-Fuzzy Classifying and Neuro-Fuzzy Clustering meet four of the five **Must** requirements; it has been shown to be as accurate as standard neural network IDSs even in early testing and shows improvement to false positive rates; it can scale to a network of any size, although it can take longer to train on large networks; it gives a clear natural language representation of which input features are considered for each classification; however, while it can be used to better predict the output this information is difficult to extrapolate and requires manual work to evaluat if the if-then statements are satisfied.	This method satisfies three of the four **Should** requirements as it gives a numerical value when scoring the contribution of different input features. The natural language if-then statements make it easy to understand the reasons for the classification and also make it clear how a classification would change if the input data was altered, though it would be difficult to manually work out for input data with greater dimensionality. However, it is unable to provide examples of inputs which would give specific classifications.	Currently none of the **Could** requirements are satisfied. It is unable to generate responses to questions from a human. Information about detections could be extrapolated from the if-then statements and displayed on a GUI, but this would require further work. It would be possible to work out ranges of results which would cause specific classifications, however this would require a lot of additional work.	This method currently has no preventative systems, meaning it does not meet the **Would** requirement, however information extrapolated from the if-then statements would make deciding on preventative measures easier for an IPS.

Adversarial Pair of Networks	This method meets all five of the **Must** requirements. As it can be applied to any other neural network IDS, it is as accurate as the current best system with as low false of a positive rate and can be used to gain knowledge on how to improve those systems. It scales to a larger network as well as the system it is applied to. It gives clear natural language statements showing exactly which input features contributed to a classification and why. It also gives a clear picture of how inputs correlate to outputs, making it easier to predict the output classification.	This method meets two of the four **Should** requirements. It gives the best explanations for classifications with in-depth descriptions of the input data and shows how the classification would change if the input data was altered. It does not currently show the importance of each input feature for each classification type however this information could be extrapolated manually, or automatically in future works. This method could also be adapted to allow the output of example input data to create a specific output, but this is not currently implemented.	Currently none of the **Could** requirements are satisfied. This method could be adapted to allow for questioning the network with simple queries about detection, by having it manipulate the flagged input data and seeing the result, but this is not currently implemented. The natural language statements would make a GUI far more understandable, with a clear description of each detection, allowing an expert to quickly decide how to react, however this method is still in early testing and is not currently implemented into	This method currently has no preventative systems, meaning it does not meet the **Would** requirement, however information extrapolated from the statements would make deciding on preventative measures easier for an IPS.

			a finished product. The adversarial network could be altered to test different possibilities of input data to determine a model of how different ranges of inputs can lead to specific outputs, but this is not yet implemented.	

Layer-Wise Relevance Propagation	This method meets three of the five **Must** requirements. As it can be applied to any other neural network IDS it is as accurate as the current best system with as low false positive. It gives a clear picture of which input features are important for a classification, with the added advantage of showing hidden layer nodes contributions as well. It partially aids in the ability to predict outputs, but this would require a lot of work tracing how values would change as they propagate through the network. While it can scale to a network of any size and feature dimensionality, it will slow considerably with current algorithms.	This method satisfies one of the four **Should** requirements. It clearly shows the degree of contribution from each input feature; however, it is unable to give any explanations for how these input features are relevant or how changing the results might change the classification. While it can give a rough idea as to which input values could result in a specific classification, it is not detailed or accurate enough to be of much use.	This method meets none of the **Could** requirements. It is unable to generate any explanations for input features or reasons for desertions made by the network. It provides little use for aiding a GUI and can only provide a general idea as to how different inputs would propagate through the network, and not any specific values or ranges.	Layer-wise relevance propagation does not meet the **Would** requirement and provides little aid for a future IPS.

4.1 RESULTS

Table 2: The number of requirements from each prioritisation of MoSCoW assessment satisfied by each solution

	Number of Requirements Satisfied			
Method	**Must**	**Should**	**Could**	**Would**
Self-Organising maps	3	1	0	0
Neuro-Fuzzy Classification	4	3	0	0
Neuro-Fuzzy Clustering	4	3	0	0
Adversarial Pair of Networks	5	2	0	0
Layer-Wise Relevance Propagation	3	1	0	0

The data in table 2 gives a good representation of how each solution compares when assessed against my requirements for interpretability. It shows that while each method has some positives, the methods most accurately matching my criteria is the method utilising an adversarial network trained to give explanations for the first's classifications.

4.2 Analysis

This research has shown that each of these solutions presents valid methods for intrusion detection with accuracy comparable to currently used systems, and while there is not yet a good solution for explainability in this field, some aspects of these methods have made progress towards interpretability. This shows the black-box problem to be unsolved, and still requires significant work in future to achieve an acceptable solution.

The best method I have analysed for aiding interpretability in IDSs is using an adversarial network. The simple natural language statements produced based on why each individual data was classified the way it was is a great improvement over any method showing only which input features contributed to different classification types and provides clear insight into the cause effect relationship between input and output. The statements would make it clear if an attack was a genuine threat or false positive to any experienced user and give a good idea how the attack could be countered.

The method utilising layer-wise relevance propagation had the advantage of showing the internal structure of the network, however it fell short in its ability to give explanations for how the data was relevant to the classification. It was only able to produce a model for how data propagated through the network for a specific classification type, rather than showing this for each data. It required additional manual work to achieve its results, although automated solutions are possible.

The methods Neuro-Fuzzy Classification and Clustering were both able to produce information which aided the prediction of results by a human, however this information had to be manually extrapolated from the if-then statements to be useful. This information was almost as good as the statements from an adversarial network, however it lacked the ability to show this for each data and could only show what inputs were contributing to each classification type.

Finally, the visual representation of the input data generated by the self-organising maps gave a useful insight into the patterns and structure of the data, with clear demonstration of how input features were related. While this is useful for a classifier and could potentially make a network more accurate, it does not give a lot of information aiding interpretability. It is useful for understanding the data and can help a human to appreciate how a neural network might view input features, however it does not contribute to solving the black-box issue.

From this work, I conclude that the best method for improving interpretability in a neural network-based network intrusion detection system is to use a combination of

three of the proposed methods I have shown. Including a self-organising map or kohonen layer to reduce the dimensionality and complexity of the input data can help with accuracy and pattern recognition as well as be a useful aid for a human to extrapolate information from. Training a second network as an adversary designed to give explanations for each classification based on how the network reacts to changes in the input data generates natural language statements which greatly aid a human's ability to predict the results and shows the relationship between input output correlation and is the best single method currently for improving interpretability. Additionally, using layer-wise relevance propagation to analyse the internal structure of the network can give even greater understanding to how the network classifies data, and helps to alleviate some of the unknown factors of the structure.

5 CONCLUSIONS

In this paper I have shown a number of methods which strive for a greater level of interpretability and explainability in the field of neural network-driven network intrusion detection systems. While none of the proposed methods fully encompass the requirements I set, some were successful in part. From this work I have concluded that an adversarial approach is most suitable for generating reliable explanations for classifications made by an NN-IDS. The ability to be applied to any neural network type is advantageous and allows it to stay on par with the current best systems, and its natural language statements showing exactly which input features contributed to each classification were easily understandable and would give useful insight to a human when a potential detection is made. Based on my definitions I believe this method qualifies as interpretable but falls short on being explainable. The adversary can consistently give insights into which input features led to specific classifications but is unable to explain the logic and reasoning behind them. From this I conclude that the best solution for creating an interpretable intrusion detection system utilising neural networks is to use a combination of an adversarial network trained to test the reasons for each detection, include a self-organising map or kohonen layer to simplify input data features into a 2 dimensional space, aiding both pattern recognition and accuracy, and giving a human the ability to view features of the data in a better form, and to analyse the network with layer-wise relevance propagation to give a better understanding of the internal structure of the network.

In summary, this work has shown that a great deal more research and development is needed in this field before the black-box issue can be solved, and until it is, we will have to be sceptical of any results produced by a neural network. This does not diminish the usefulness of the technology; however, it does hinder its progress and adoption into modern society and can add risk to using it in intrusion detection, especially applied to critical infrastructure. Without a greater level of understanding as to the deeper knowledge gained by the network, we will be unable to predict how it will react to all situations. If an IDS built upon these technologies was to receive data in a form it has not trained with, or been tested against, it could react in a way which causes harm to the system, either by allowing an attack onto the system or by recommending defensive measures for normal data.

APPENDIX E – DRIVE CONTENTS

Attached USB drive contains:

- Project report