Comparison of a Support Vector Machine and a Artificial Neural Network for Automatic Modulation Recognition using Higher Order Cumulants

Jaimeen Lalloo, 1855116

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: **Adaptive Modulation and Coding adjusts the modulation order based on the channel conditions to ensure reliable data transfer. Automatic Modulation Recognition (AMR) is the process of identifying the modulation format at the receiver. AMR is used in conjunction with AMC to allow users to receive data at low Signal to Noise Ratios. The project aims to create a blind AMR system which can effectively recognize a modulation scheme at low SNR. An indoor channel environment is created using the IEEE802.11a standards, which include AWGN and a Rayleigh fading channel. The modulation schemes tested are: BPSK, QPSK, 16QAM and 64QAM. The system first extracts features, and the features are then fed to a Machine Learning classifier. A Support Vector Machine (SVM) classifier is compared to a Neural Network (NN) Classifier. The parameters of the NN are optimized to find the most accurate NN. A wide neural network with 100 node and a single layer is the most effective with a highest accuracy of 85.6% in an AWGN environment at 10dB SNR and a lowest accuracy of 44.8% in a fading environment at -10dB SNR. The performance of the SVM exceeds that of the NN. A Fine Gaussian SVM achieves a highest accuracy of 94.7% in an AWGN environment and a lowest accuracy of 85.2% in a fading environment at -10dB. The performance of the SVM at low SNR is extremely efficient due to the higher dimension mapping and is seen as a solution to increase the range of WI-FI routers and cellular towers. Further recommendations include using the classifier output and a channel equalizer to allow for demodulation of the received signal.**

Table of Contents

1. INTRODUCTION

Adaptive Modulation and Coding (AMC) is used to accommodate for dynamic channel conditions. Initially used in military applications, AMC has migrated to general wireless communications and are part of WI-FI standards such as IEEE 802.11n and IEEE 802.11ac [1]. AMC adjusts the modulation order or the coding rate based on the channel conditions to ensure reliable data transfer. The modulation type is a prerequisite for demodulation at a receiver. However, in AMC, the modulation is constantly changing based on the channel conditions. This allows data transmission at lower SNR.

Automatic Modulation Recognition (AMR) is the process of identifying the modulation format at the receiver. Machine. AMR is used in conjunction with AMC to allow users to receive data at low SNR. The aim of the project is to create a blind AMR system which can effectively recognize a modulation scheme at low SNR. Applications of the system include expanding the range of WI-FI routers and cellular towers.

2. BACKGROUND

Initial AMR systems consisted of model-based calculations based on link-quality metrics [2]. However, complex models and low accuracy deemed the system unacceptable. Future models for AMR were split in two different approaches: Likelihood-Based (LB) recognition and Feature-Based (FB) extraction (section C).

In addition to the extracted features, Machine Learning (section D) was introduced into the model. This largely improved the accuracy.

The environment in which the system will be tested is that of a WI-FI router (IEEE802.11a standard) in an indoor environment. The channel will consist of Additive White Gaussian Noise (AWGN) and will have Rayleigh fading. The modulation schemes classified include: BPSK, QPSK, 16QAM and 64QAM.

Signal Classification consists of extracting features from a received signal, and feeding the features to a Machine Learning classifier as shown in figure 1.

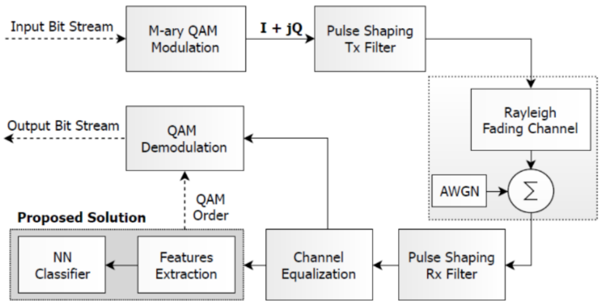

Figure 1: Automatic Modulation Recognition

A. Modulation Schemes

1) Phase Shift Keying(PSK)

PSK modulates a signal by varying the phase of the carrier signal. The modulated signal is given by the following equation where p(t) is a complex value representing the baseband signal.

$$u_m(t) = p(t)\cos(2\pi f_c t + \varphi_m) \qquad (1)$$

$$\varphi_m = \frac{2\pi m}{M} \quad m = 0,1 \ldots M-1 \qquad (2)$$

$$u_m(t) = g(t)A_{mc}\cos(2\pi f_c t) - g(t)A_{mc}sin(2\pi f_c t) \ (3)$$

In Phase term

$$A_{mc} = \cos\left(\frac{2\pi m}{M}\right) \qquad (4)$$

Quadrature term

$$A_{ms} = \sin\left(\frac{2\pi m}{M}\right) \qquad (5)$$

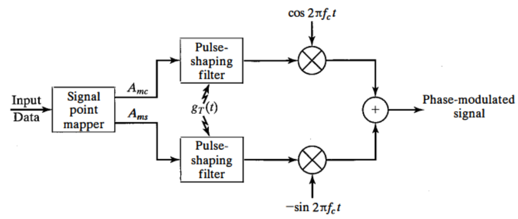

Figure 2: Phase Shift Keying Modulator Block Diagram [3]

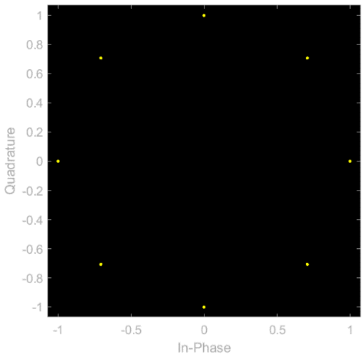

Figure 3: Constellation Diagram Phase Shift Keying

A PSK signal can be demodulated using either a Phase Lock Loop Demodulator or a Decision Feedback Demodulator.

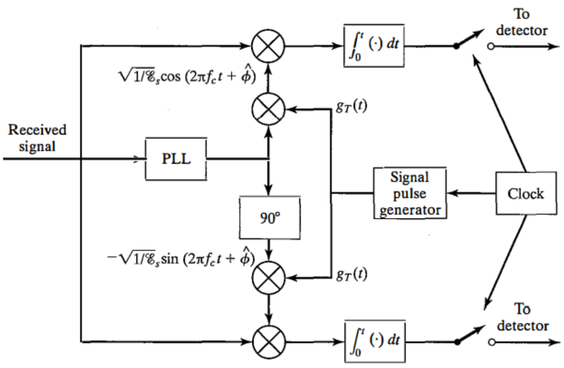

Figure 4: PLL Demodulator for PSK [3]

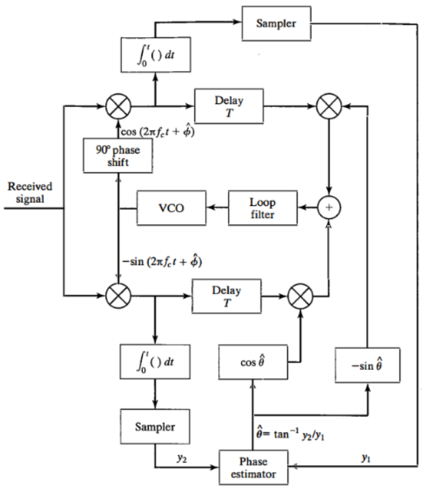

Figure 5: Decision Feedback Demodulator for PSK [3]

During demodulation, the signal can be misclassified based on the distance between the points on the constellation diagram. The figure below shows the Probability of Error for the demodulation of MPSK signals.

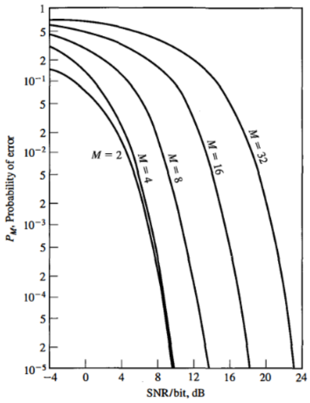

Figure 6: Probability of error in demodulation for MPSK [3]

2) *Quadrature Amplitude Modulation(QAM)*

QAM is a combination of digital amplitude modulation and digital phase modulation.

$$u_m(t) = A_{mc} g_T(t)\cos(2\pi f_c t) - A_{ms} g_T(t)\sin(2\pi f_c t) \quad (6)$$

Figure 7 shows a block diagram of a QAM modulator.

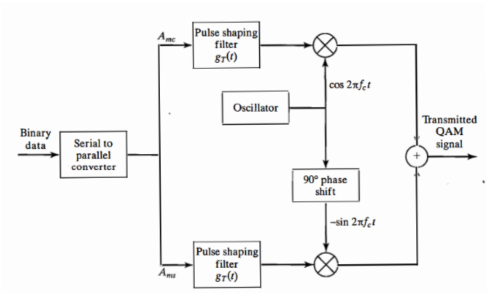

Figure 7: MQAM Modulator Block Diagram [3]

The figure below shows the block diagram of a QAM demodulator.

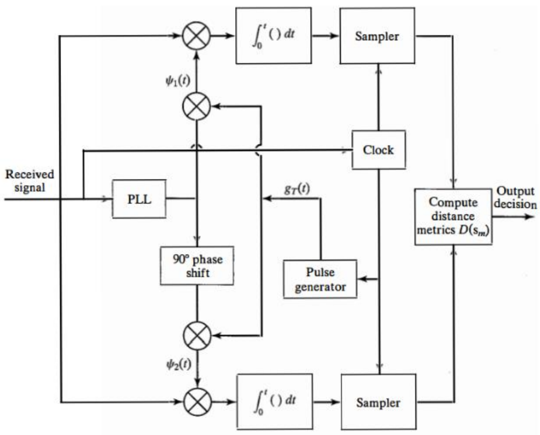

Figure 8: MQAM Demodulator [3]

The figure below shows the probability of error in the demodulation of a signal.

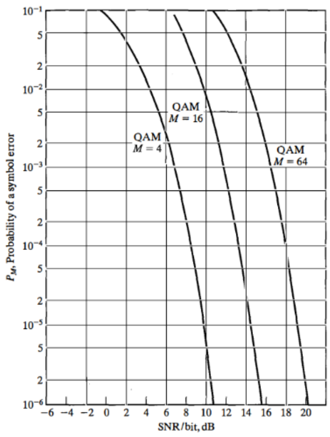

Figure 9: Probability of Error for MQAM Demodulation [3]

B. *Channel*

1) *Fading Channel*

A fading channel can be either frequency selective or frequency non-selective. Frequency selective is when the symbol duration is shorter than the path delay. This results in the received signal being shown in the equation 7.

$$r(t) = \sum_{n} \alpha_n(t) e^{-j2\pi f_c \tau_n(t)} \qquad (7)$$

Where $\alpha_n(t)$ is the time variant attenuation factor associated with the n^{th} propagation path and $\tau_n(t)$ is the propagation delay. $\alpha_n(t)$ is described by the Rayleigh Probability Distribution(f(x)), where x is proportional to the path loss in dB and μ is the mean path loss in dB and σ is the standard deviation.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (8)$$

2) *Channel Equilization*

Channel Equilization aims to compensate the distortion caused by time-variant channels. Channel Equilization consists of two algorithms:

- ZERO-FORCING
- LMS

C. *Features*

1) *Likelihood Based(LB)*

A probabilistic model is created for the received signal by applying a Likelihood Function (LF) to the received signal. This LF is then compared to a Bayesian criterion threshold to determine the modulation type [5]. The LF and Bayesian criterion is used to generate a Likelihood Ratio Test (LRT). Thereafter, the Maximum Likelihood Estimations (MLE) are derived from the LRTs. Further improvements include a hybrid likelihood ratio test (HLRT) and a quasi HLRT [6].

The MLE AND HLRT approach results in high accuracy, however it also has a high computational complexity and is sensitive to noise, phase offsets and frequency offsets. The QHLRT is less complex, however, the model does not perform well under low SNR conditions.

2) *Feature Based (FB)*

Features are extracted from the received signal. The features are processed to determine the type of classification used and thereafter the received signal is demodulated. There are four types of features which are extracted from a signal, namely: Spectral, Statistical, Transform and Constellation shape.

a) *Spectral Features*

Spectral features utilize primary features to extract several secondary features from a received signal.

The primary features include instantaneous amplitude ($A(t)$), phase ($\varphi(t)$) and frequency (f_N). The most common method to extract the primary features include using a Hilbert Transform [2]. Equation 1-3 shows the derivation of primary features, where r is the received signal and $\mathscr{r}$ is the Hilbert Transform of the received signal. The full list of secondary features is shown in figure 10.

$$A(t) = \sqrt{r^2(t) + \mathscr{r}^2(t)} \qquad (12)$$

$$\varphi(t) = \tan^{-1}\frac{\mathscr{r}(t)}{r(t)} \qquad (13)$$

$$f_N = \frac{d_{\varphi uw}(t)}{2\pi dt} \qquad (14)$$

Features	Mathematical Equation
Maximum value of PSD γ_{max} of the normalized centered instantaneous amplitude	$\gamma_{\max} = \frac{\max\lvert DFT(A_{cn}(i)\rvert^2}{N_s}$, where DFT is the discrete Fourier transform of the modulated signal, N_s is the sample number, $A_{cn} = \frac{A_i}{\mu_A} - 1$, A_i is the i^{th} instantaneous amplitude and μ_A is the sample mean
Standard deviation of the absolute values of the centered nonlinear components of instantaneous phase σ_{ap}	$\sigma_{ap} = \sqrt{\frac{1}{N_c}\left(\sum_{A_n(i)>A_t} \varphi_{NL}^2(i)\right) - \frac{1}{N_c}\left(\sum_{A_n(i)>A_t} \lvert\varphi_{NL}(i)\rvert\right)^2}$, where N_c is the number of sample(s) in $\{\phi_{NL}\}$ for $A_n\ (i) > A_t$, where A_t is the threshold value of $A_n\ (i)$ when the filter provides the minimum amplitude of the signal sample due to high noise sensitivity and $\phi_{NL}\ (i)$ is the nonlinear component of the i^{th} instantaneous phase of the sample.
Standard deviation of the absolute value of the normalized centered instantaneous amplitude in the nonweak segment of signal σ_a	$\sigma_a = \sqrt{\frac{1}{L}\left(\sum_{A_n(i)>t_{th}} a_{cn}^2(i)\right) - \frac{1}{L}\left(\sum_{A_n(i)>t_{th}} \varphi_{cn}(i)\right)^2}$, where L is the length of the non-weak value and t_{th} is the threshold value of the non-weak signal.
Standard deviation of the direct value of the centered nonlinear component of the direct instantaneous phase in nonweak segment σ_{dp}	$\sigma_{dp} = \sqrt{\frac{1}{N_c}\left(\sum_{A_n(i)>A_t} \varphi_{NL}^2(i)\right) - \frac{1}{N_c}\left(\sum_{A_n(i)>A_t} \varphi_{NL}(i)\right)^2}$, where all parameters are similar to σ_{ap} but differs in the absence of the absolute operator in the nonlinear component of the instantaneous phase.
Standard deviation of the absolute value of the normalized centered instantaneous amplitude of signal segment σ_{aa}	$\sigma_{aa} = \sqrt{\frac{1}{N_c}\left(\sum_{i=1}^{N} A_{cn}^2(i)\right) - \frac{1}{N_c}\left(\sum_{i=1}^{N} \lvert A_{cn}^2(i)\rvert\right)^2}$, where A_{cn} is the normalized and centered instantaneous amplitude of the incoming signal at the time instant.
Standard deviation of the absolute value of the normalized centered instantaneous frequency of signal segment σ_{af}	$\sigma_{af} = \sqrt{\frac{1}{N_c}\left(\sum_{A_n(t)>A_t} f_N^2(i)\right) - \frac{1}{N_c}\left(\sum_{A_n(t)>A_t} \lvert f_N(i)\rvert\right)^2}$, where f_N is the normalized frequency.
Kurtosis of the normalized centered instantaneous amplitude μ_{42}^a	$\mu_{42}^a = \frac{E\{A_{cn}^4[n]\}}{\{E\{A_{cn}^2[n]\}\}^2}$.
Kurtosis of the normalized centered instantaneous frequency μ_{42}^f	$\mu_{42}^f = \frac{E\{f_N^4[n]\}}{\{E\{f_N^2[n]\}\}^2}$.

Figure 10: List of Spectral Features which can be extracted from a signal [4]

The features have different properties based on the type of modulation. An example is μ_{42}^{α} illustrates the deviation of the amplitude and the frequency. This can be used to distinguish between PSK and QAM. Spectral Feature extraction reduces the complexity of the receiver and computational time, however, the method is not resistant to noise [7] and it does not perpetuate efficiently to higher order modulation schemes [8].

b) *Statistical Features*

Statistical features implement Higher-Order Statistics to extract features from a signal. Cumulants were first introduced in 1889 by Thorvald N. Thiele. Moments and Cumulants are calculated from the received signal and used to determine the shape of a signal. The equation for calculating the moment of a received signal is given by the equation below.

$$M_{pq} = E[y(k)^{p-q} y^*(k)^q]$$

The cumulants of a signal can be determined using various moments of a signal. This is shown in table 1.

Table 1 : List of Cumulants which can be extracted from a signal [9]

Cumulant	Equation
C_{20}	$E[y^2(n)]$
C_{21}	$E[\lvert y(n)\rvert^2]$
C_{40}	$M_{40} - 3{M_{20}}^2$
C_{41}	$M_{40} - 3M_{20}M_{21}$
C_{42}	$M_{42} - \lvert M_{20}\rvert^2 - 2M_{21}$
C_{60}	$M_{60} - 15M_{20}M_{40} + 30{M_{40}}^2$
C_{61}	$M_{61} - 5M_{21}M_{40} - 10M_{20M_{41}} + 30{M_{20}}^2M_{21}$
C_{62}	$M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6{M_{20}}^2M_{22} + 24{M_{21}}^2M_{22}$
C_{63}	$M_{63} - 9M_{21}M_{42} + 12{M_{21}}^3 - 3M_{20}M_{43} - 3M_{22}M_{41} + 18M_{20}M_{21}M_{22}$

[9] identifies HOCs as a more effective method as compared to Spectral features in classifying signal in fading channels.

c) *Transfom Features*

Wavelet Transform(WT) analyses a signal at a different frequencies and resolutions. The Continuous Wavelet Transform of a received signal is given by equation 15.

$$CWT(\alpha,\tau) = \frac{1}{|\alpha|}\int_{-\infty}^{\infty} S(t)\varphi^{*}{}_{\alpha,\tau}\left(\frac{t-\tau}{\alpha}\right)dt \quad (15)$$

WT is combined with a Convolutional Neural Network to achieve accurate classification, however, the receiver is highly complex [10].

d) *Constellation Shape Features*

The method uses the distance and the difference in phase from the origin point to identify the modulation scheme.

D. *Machine Learning*

Machine Learning(ML) consists of two categories; Supervised and Unsupervised. Supervised ML is further comprised of Regression and Classification, whereas unsupervised ML consists of Clustering and Reinforcement Learning.

1) *Supervised Machine Learning*

a) *K Nearest Neighbour(KNN)*

The algorithm searches for K of its nearest neighbours(points surrounding it) and uses the distance between the point and its neighbours to classify the data point. Distances can be calculated using Euclidean, Manhattan or Minkowski distance. The advantage of KNN is that it is easy to implement, however, the computation time is large.

b) *Naïve Bayes*

A Naïve Bayes classifier uses probability theory and the Bayes Theorem to classify data points. The solution works well with small training set, however, the input features are required to be independent of each other.

c) *Decision Trees(DT)*

Threshold levels are decided at each branch of the tree. Recursive binary splits occur and a Gini score is calculated to determine the effectiveness of the split. The Gini score is calculated using a cost function. A cost function is the difference between the predicted outcome and the actual outcome. DT allow for the mapping of complex data and has a small computational time. [11] compares a SVM to a DT and shows that the DT is highly susceptible to overfitting.

d) *Linear Regression*

Linear Regression aims to determine a linear relationship between the outcome and the inputs. The model starts with a arbitrary model and then tweaks the model based on the cost function and gradient descent. Gradient Descent is finding the gradient of the cost function and moving in a direction which minimizes the cost function. The advantage of Linear Regression is that it is fast and easy to implement, however, the solution does not fit well with complex input data.

e) *Support Vector Machines*

SVMs aim to not only create a boundary between the classes, but also to maximize the margin between the boundary and the classes.

$$x^T = [x_1\ x_2\ \dots\ x_n\ 1] \qquad (16)$$

SVMs are also composed of a linear discriminant vector.

$$w^T = [w_1\ w_2\ \dots\ w_n\ b] \qquad (17)$$

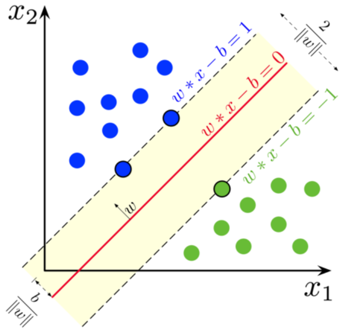

Figure 11: Binary Support Vector Machine Classifier

Features (x_n) on the line $x^T w^T - b = \pm 1$ are referred to as Support Vectors. The distance between a support vector and the boundary is referred to as the margin (m), where x_1 and x_2 refer to support vectors from different classes.

SVMs consist of a feature space. The feature space can take the input features and map it to a higher dimension to allow the machine to construct nonlinear boundaries. The input features are mapped to higher dimensions using $\emptyset(x)$.

The margin of the SVM is given by an inference function (h(x)).

$$h(x) = w^T\emptyset(x) + b \qquad (18)$$

A discriminant function g(z) is applied to the inference function g(h(x)). The discriminant function is the sign function which allows for binary classification of the input features as either a 1 or -1 as shown in the figure below. SVMs can only classify between two classes at a time, therefore, µ SVMs are required to classify µ classes.

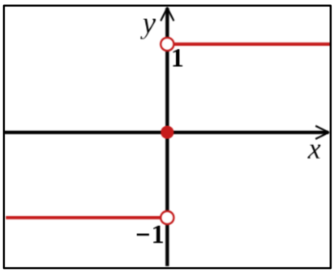

Figure 12: sign(x) function

The boundary is given by equation 19, where b is the margin bias.

$$x^Tw^T - b = 0 \qquad (19)$$

The functional margin ($\hat{y}_n$) compares the predicted class to the actual class and the equation is shown below, where y_n is the actual class.

$$\hat{y}_n = y_n(w^T\emptyset(x) + b) \qquad (20)$$

The geometric margin (γ_n) is the perpendicular distance between a point and the decision boundary. Equation 21 shows the geometric margin, where x_n is the input:

$$\gamma_n = \gamma_n\left(\left(\frac{w^T}{\| w \|}\right)\emptyset(x_n) + \frac{b}{\| w \|}\right) \qquad (21)$$

The margin for the entire set is the distance between the closest support vector and the boundary. Therefore, minimizing $\| w \|$ for the smallest margin will maximize the margin. W can also be written as shown in equation 22.

$$w = \sum_{n=0}^{N-1} \alpha_n y_i x_i \qquad (22)$$

The maximization problem can be written as equation 23 (the derivation is shown in the engineering notebook)

$$\sum_{n=0}^{N-1} \alpha_n - \frac{1}{2}\sum_{n,n'=0}^{N-1} \alpha_n\alpha_{n'}\gamma_n\gamma_{n'}{x_n}^T x_{n'} \qquad (23)$$

Where α determines whether the input is a support vector or not. The aim of the optimization problem is to maximize α and this is done by gradient descent. ${x_n}^T x_{n'}$ is a measure of the similarity of the vectors and can be defined as a kernel function ($\kappa(x_n, x_{n'})$).

$$\kappa(x_n, x_{n'}) = {x_n}^T x_{n'} = \emptyset(x_n)^T\emptyset(x_{n'}) \qquad (24)$$

Computing $\emptyset(x_n)$ can be computationally expensive and Kernels help reducing the computation by calculating $\kappa(x_n, x_{n'})$ without mapping it to higher dimensions. The kernels also map out non-linear boundaries.

Popular kernels include:

- Linear Kernel

$$\kappa(x_n, x_{n'}) = {x_n}^T x_{n'} \qquad (25)$$

- Polynomial Kernel
- Where d is the degree of the polynomial.

$$\kappa(x_n, x_{n'}) = \left({x_n}^T x_{n'}\right)^d \qquad (26)$$

- Gaussian Kernel

$$\kappa(x_n, x_{n'}) = e^{-\gamma\|x_n - x_{n'}\|^2} \qquad (27)$$

The advantage of SVMs is that the model is able to use a kernel function to map complex input data. The algorithm also has a convex optimization function, which ensures it will always reach the global minima and not a local minima. However, larger training times are required for larger datasets and multiple SVMs are required for multiclass classification.

The computational complexity of a SVM is O(μ*N), where N is the number of input features and μ is the number of output classes. The space complexity is O(μ*N^2), due to the Kernel function.

f) Neural Networks

The first neural network was introduced in 1957 by Frank Rosenblatt. The network consisted of a Perceptron.

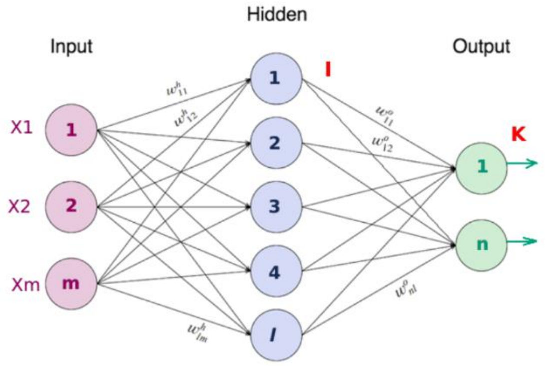

Figure 13: Single Layer Neural Network

A neural network consists of three parts: the input layer, the hidden layers and the output layer. The number of input nodes correlate to the number of input features (x^T).

$$x^T = [x_1\ x_2\ ...\ x_n] \quad (28)$$

The hidden layers consist of neurons/nodes. In a fully-connected network, each node from the input layer is connected to all the nodes in the second/hidden layer with weighting(w).

$$w^T = [w_1\ w_2\ ...\ w_m] \qquad (29)$$

The values of the all the nodes in the previous layer are multiplied by the weighting and summed up. This value is then passed to an Activation Function to derive the value of the node (h_i).

$$h_i = \emptyset(\sum_{j=1}^{m} w_{i,j}x_j) \qquad (30)$$

Activation Functions are considered as hyper-parameters and can be changed based on the problem. Popular Activation Functions include: Rectified Linear Unit, Sigmoid and the Softmax Function.

In a classification problem, the number of output nodes depend on the number of classes. The parameters of the hidden layers can be adjusted based on the problem. The parameters include the number of hidden layers and the number of neurons in each hidden layer. Research done on parameter optimization resulted in the Universal Approximation theory, which states that three hidden layers can approximate any function given enough nodes and the right weights.
A neural network consists of a training stage, in which the network receives labelled data and through the process of Backpropagation, the network adjusts its weights to improve accuracy. The error is determined by a loss function. Common loss functions include Mean Square Error (MSE) and Binary Cross Entropy (BCE). The loss function determines the difference between the predicted class($ð_{i,q}$) and the actual class($d_{i,q}$). The Mean Square Error Loss function is shown in equation 31.

$$f = \sum_{i=1}^{N}\sum_{q=1}^{Q}\frac{1}{2}(ð_{i,q} - d_{i,q})^2 \qquad (31)$$

This results in an optimization problem, which requires the minimization of the loss function. The optimization problem requires finding the derivative of f and minimizing the error by moving toward a negative gradient. This is shown in equation , where α_t is the learning rate and is a hyper-parameter which can be altered. The full derivation of the equation is shown in the engineering notebook.

$$w_{k,l}{}^{t+1} = w_{k,l}^{t} - \alpha_t \delta_k^i h_l^i \qquad (32)$$

$$\delta_k^i = (ð_k{}^i - d_k{}^i)ð_k{}^i(1 - ð_k{}^i) \quad (33)$$

The computational complexity for a Neural Network O(n*t*(i*j+j*k+k*l)), where n is the number of training samples and t is the number of epochs; i, j, k, l is the number of nodes in the first, second, third and output layer.

[12] implements an Artificial Neural Network (ANN) with all the Spectral features as inputs and compares it to a Deep Neural Network (DNN).

The ANN has an accuracy of 65% in a fading environment, whereas the DNN has an accuracy of 100%. However, the DNN consists of 700 nodes in total and has a large computation time. Therefore, the DNN is not seen as a feasible solution.

3. Literature Review

[4] is a summary of studies done in the field of Automatic Modulation Recognition. The study compares the features mentioned before and ML techniques. According to the study, HOCs is currently the most deterministic feature in AMR. The most efficient machine learning tools include Support Vector Machines (SVM) and Artificial Neural Networks (ANN). Therefore, the project aims to compare the SVMs to ANNs using HOCs as an input.

4. Implementation

A. *Environment*

An HP laptop with a Ryzen 5 4500U processor and Radeon Graphics is used to develop the project. The laptop has 8GB of RAM and has a Windows Operating System.
MatlabR2021a is used as the development environment as compared to Python with Tensorflow due to the virtual environment (Anaconda) requirement. The GPU acceleration also require Nvidea GPUs and is not compatible with AMD GPUs. A secondary reason for using the Matlab software is the toolboxes which Matlab provide, such as the Communications Toolbox and Classification Learner.

B. *Data*

The modulation schemes being investigated are digital. Therefore, the data points are generated as a random string of binary bits. The bits are then modulated according to the modulation schemes in the Communications Toolbox. An example of the modulation schemes is the *pskmod(data, M)* function. The function implements M-Ary Modulation on the string of binary bits and returns complex values which represent the position of the signal on a constellation diagram.

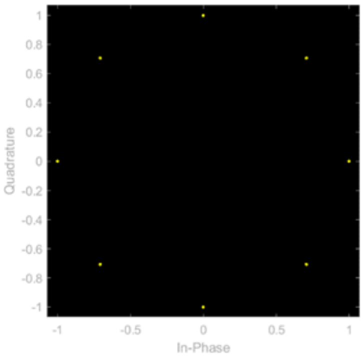

Figure 14: Constellation Diagram of an 8PSK Signal with no noise

The signal is assumed to be transmitted at baseband, which is a vector of complex value signals (txSig).

C. *Channel*

1) *Additive White Gaussian Noise*

Additive White Gaussian Noise is added to the signal using the awgn(txSig, SNR) function is the Communications Toolbox. The SNR values measured range from -5dB to 10dB. The figure below shows the received signal with an SNR of 5dB.

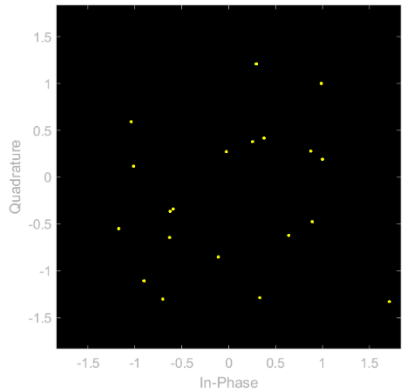

Figure 15: Constellation Diagram of an 8PSK signal with 5dB SNR

2) *Rayleigh Channel*

The Rayleigh Channel is implemented using the comm.RayleighChannel() function. The channel has a sample rate of 1000Hz. The channel is a multipath channel with a line-of-sight path and two reflected paths. The first reflected path consists of a path delay of 0.001s with a gain of -3dB and the second path has a delay of 0.002s with a gain of -6dB. AWGN is added to the signal. The figure below shows the fading channel with a SNR of 5dB.

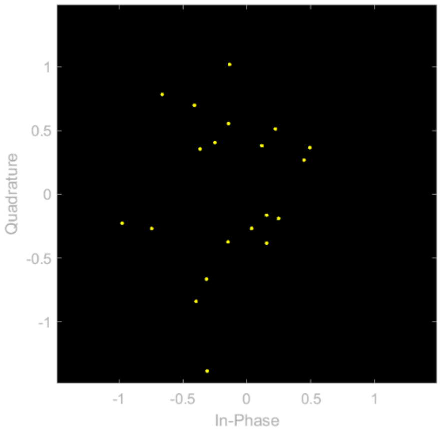

Figure 16: Constellation Diagram of a 8PSK signal in a Fading Channel with 5dB SNR

D. Receiver

1) Preprocessing

The output of the channel is stored as a matrix. The size of each row represents the frame size, ie, the size of the received signal. The number of rows represent the number of frames, ie, the number of received signals with which we want to train/test the receiver.
The received signal is preprocessed by sending it to a function called get_cumulants(), which extracts the statistical features from the received signal. The features are stored in memory and then fed to a classifier. The figure below shows a scatterplot of the first two cumulants.

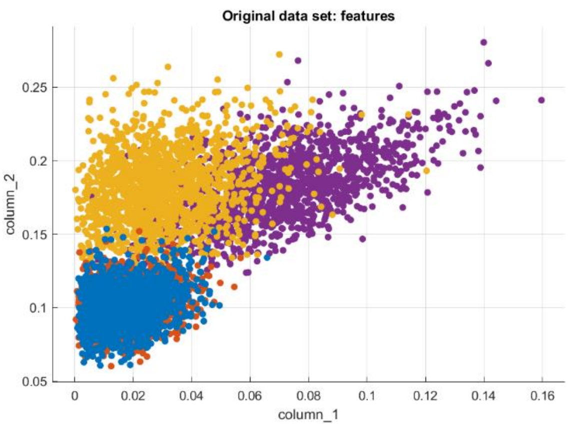

Figure 17: Scatterplot showing the first two cumulants and the corresponding modulation in a fading channel at 10Db SNR

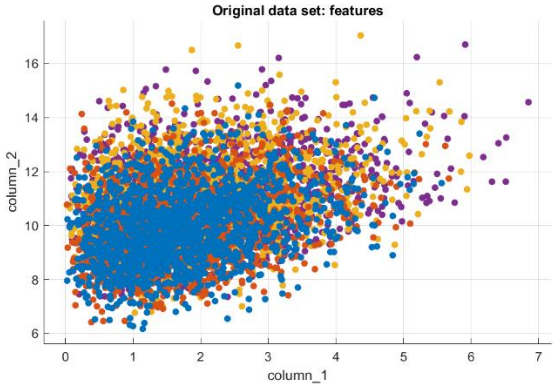

Figure 18: Scatterplot showing the first two cumulants and the corresponding modulation in a fading channel at -10Db SNR

2) Classification

The aim of the project is to compare a Support Vector Machine Classifier to a Neural Network Classifier.

The parameters of the Neural network are investigated to determine the most effective Neural Network.

Neural Network Parameters

1. Number of Nodes
2. Number of Hidden Layers

Neural Network	Nodes in each layer	Number of Layers
Narrow	10	1
Medium	25	1
Wide	100	1
Bilayered	10	2
Trilayered	12	3

The Neural networks are tested in both AWGN environments and Fading environments at five different SNRs (-10, -5, 0, 5) and the accuracy is recorded. The size of the received signal is 50 bits and the models are trained with 1500 signals per modulation.

Support Vector Machines(SVM)

SVMs have a single parameter that can be optimized, which is the Kernel function.

Kernel Functions

1. Linear
2. Fine Gaussian

The SVMs are tested in the same conditions as the Neural Networks to ensure standards conditions.

5. Results and Performance Analysis

The first result tests for an adequate size of the received signal. Previous studies have tested with a size of 256, 512, 1024 and 2048. However, this increased latency. Therefore, smaller test size were tested. The accuracy generated is that of a medium (12 nodes, 1 layer) neural network in a fading channel with -5dB. The graph shows a rapid increase inaccuracy from 2 until 50 bits, thereafter, the slope of the graph decreases. Therefore 50 bits is set as the frame size for the rest of the project.

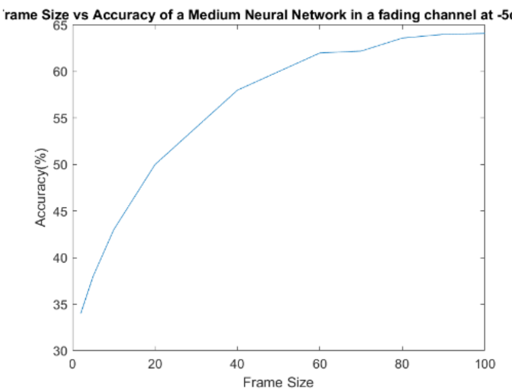

Figure 19 : Frame size vs Accuracy for a Neural Network in a Fading channel at -5dB

A. *Additive White Gaussian Noise*

1) *Neural Network*

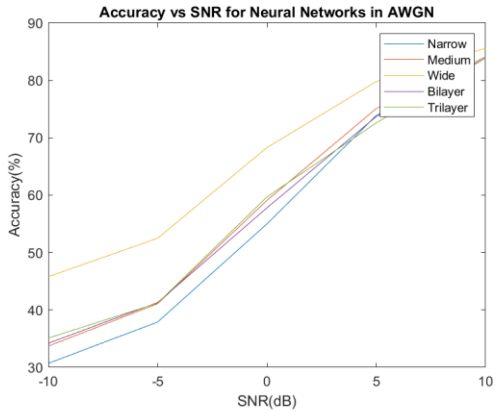

Figure 20: Accuracy vs SNR for a Neural Network in AWGN

2) *Support Vector Machine*

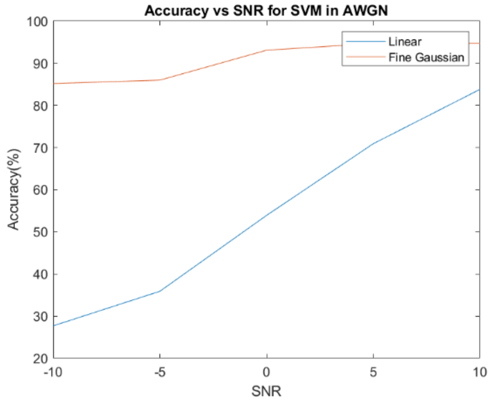

Figure 21: Accuracy vs SNR for a Support Vector Machine in AWGN

B. *Fading Channel + Additive White Gaussian Noise*

1) *Neural Network*

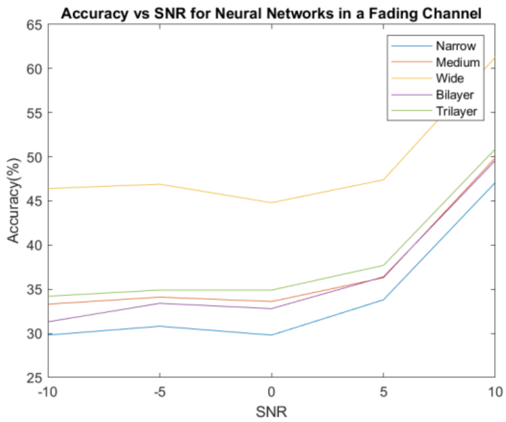

Figure 22: Accuracy vs SNR for a Neural Network in a fading channel

2) *Support Vector Machine*

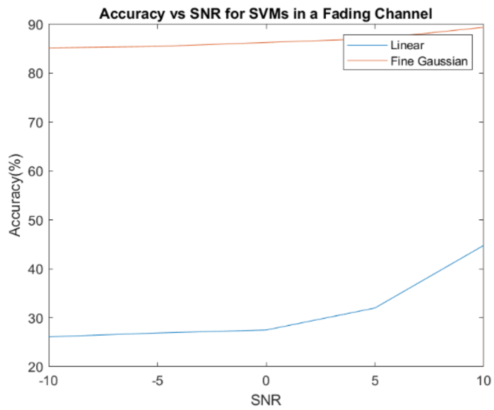

Figure 23: Accuracy vs SNR for a SVM in a fading channel

The Wide Neural Network(100 nodes) can be seen as the most accurate Neural Network. It has accuracies which are 10% higher than the rest of the models at -5dB. The wide model performs better than the rest of the models at all SNR, but the difference is maximized at lower SNRs. However, the wide model takes the longest time to train. The Bilayer and Trilayer Networks have the largest computation time, however it does not show an improvement from the Medium network.

The linear SVM shows results similar to the Medium Neural Network, however, the Fine Gaussian SVM has an accuracy of 85.2% at -10dB, whereas the Linear SVM has an accuracy of 27.7%. The Fine Gaussian SVM is extremely accurate at both high and low SNRs.

Table 2: Comparison of the most effective Neural Network to the most effective SVM in AWGN

SNR	Wide Neural Network (%)	Gaussian SVM (%)
-10	45.8	85.2
-5	52.5	86.0
0	68.3	93.1
5	79.8	94.7
10	85.6	94.7

Table 3: Comparison of the most effective Neural Network to the most effective SVM in a fading channel

SNR	Wide Neural Network (%)	Gaussian SVM (%)
-10	44.8	85.2
-5	46.4	85.5
0	46.9	87.1
5	47.4	86.3
10	83.9	89.4

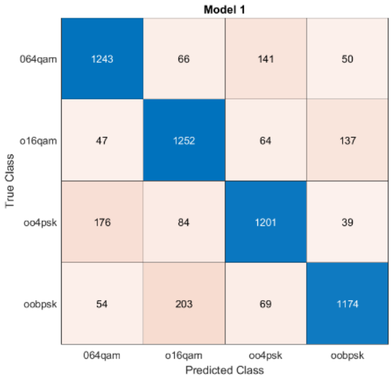

Figure 24: Confusion Matrix for the SVM at -10dB in a fading channel

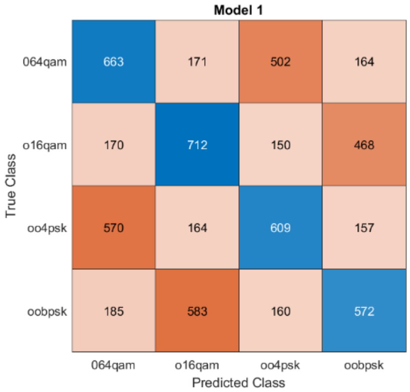

Figure 25: Confusion Matrix for the Wide Neural Network at -10dB in a fading channel

The Fine Gaussian SVM performs extremely well under all conditions, as compared to the low accuracies of the Wide Neural network at low SNR. This is attributed to the way in which the SVM operates. The mapping of the features to a higher dimension and then maximizing the margin between the boundaries allows boundaries to be extremely well defined. This is not possible with Neural Networks as they operate in a different manner. In a fading channel, the features are extremely close and cannot be separated effectively by a Neural Network. This is shown by the confusion matrices of the models at -10dB in a fading environment.

6. Social,Economical and Environmental Impact

The Digital Divide is a disparity between regions which have modern communication system and areas which do not. South Africa is a developing country, and it has a large Digital divide. The Digital Divide is seen by the availability of communication systems in between the rural and urban areas.

The social impact of the Digital Divide is that the people with no access to the infrastructure have a disadvantage. The Fourth Industrial revolution ensured digitization of processes. This includes schooling and work. However, people who cannot overcome the disadvantage of the Digital Divide will have a lack of those crucial skills, thereby increasing the inequality. Automatic Modulation and Coding is a method introduced to solve the distance problem. It allows communication over a greater distance by altering communication parameters based on the distance. This allows for cellphone towers to reach a greater distance.

Web2.0 is the term used to describe the current internet. The internet is filled with user-generated content. High internet speeds are required to keep up with the large amount of data being generated by user every moment.

Companies collect the user-generated information and use it to provide tailored services. This is known as Global Information Capitalism. High internet speeds are required to transfer the large amounts of data between businesses and clients. Faster internet speeds allow businesses to thrive on a macro scale by revolutionizing Big Data Analytics.

AMR allow for the removal of excess hardware components which results in less waste during the end-of-life process. Removing excess hardware also reduces the power consumption and carbon footprint.

The demand for internet speeds keeps rising, which makes the process economically viable. Vodacom has a vision for 2025 in which they apply Machine Learning to the telecommunications sector to keep up with the demand of internet speeds.

7. Sustainability

Growing demand for Electrical processors have put a strain on manufacturers and caused a shortage of chips. This has affected many companies, including the likes of Ford, BMW and Tesla. The shortage also affected PlayStation, with the company not being able to produce adequate supplies of its console until 2022. The chip shortage is caused by many factors including the increase in IOT devices, growth of the Cloud Computing Sector and the demand for faster telecommunications technologies(5G).

The growing demand for electrical components has resulted in a spike in E-Waste which was termed by the Wasted Electrical and Electronic Equipment(WEEE) Forum. Therefore, the telecommunications sector require technologies which can be upgraded by using a large amount of the current infrastructure. Integrating Machine Learning into base stations reduces the need for hardware such as a matched filter and a correlation detector.

Machine Learning is software dependent and therefore future upgrades to the systems may include new algorithms. The removal of intermediate hardware components reduces the energy consumption. This indirectly reduces the carbon footprint. There are less devices to manage, which results in less end-of -life management and less wasted electrical equipment.

Chip manufacturers are changing from Silicon to Gallium, which is proven to be more energy efficient. Therefore, for long-term sustainability, Gallium chips should be considered

8. Conclusion

A Support Vector Machine (SVM) classifier is compared to a Neural Network (NN) Classifier. The parameters of the NN are optimized to find the most accurate NN. A wide neural network with 100 node and a single layer is the most effective with a highest accuracy of 85.6% in an AWGN environment at 10dB SNR and a lowest accuracy of 44.8% in a fading environment at -10dB SNR. The performance of the SVM exceeds that of the NN. A Fine Gaussian SVM achieves a highest accuracy of 94.7% in an AWGN environment and a lowest accuracy of 85.2% in a fading environment at -10dB. The performance of the SVM at low SNR is extremely efficient due to the higher dimension mapping and is seen as a solution to increase the range of WI-FI routers and cellular towers. Further recommendations include using the classifier output and a channel equalizer to allow for blind signal demodulation of the received signal.

9. List of References

[1] 2011. IEEE Std 802.11v-2011, IEEE Standard for Information Technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 8: IEEE 802.11 Wireless Network Management

[2] Jensen, T., Kant, S., Wehinger, J. and Fleury, B., 2010. Fast Link Adaptation for MIMO OFDM. *IEEE Transactions on Vehicular Technology*, 59(8), pp.3766-3778.

[3] Proakis, J. and Salehi, M., n.d. *Fundamentals of communication systems*. 2nd ed. Pearson, p.256.

[4] Al-Nuaimi, D., Hashim, I., Zainal Abidin, I., Salman, L. and Mat Isa, N., 2019. Performance of Feature-Based Techniques for Automatic Digital Modulation Recognition and Classification—A Review. *Electronics*, 8(12), p.1407.

[5] J. L. Xu, W. Su and M. Zhou, "Likelihood function-based modulation classification in bandwidth-constrained sensor networks," 2010 International Conference on Networking, Sensing and Control (ICNSC), 2010, pp. 530-533, doi: 10.1109/ICNSC.2010.5461606.

[6] O. A. Dobre and F. Hameed, "Likelihood-Based Algorithms for Linear Digital Modulation Classification in Fading Channels," 2006 Canadian Conference on Electrical and Computer Engineering, 2006, pp. 1347-1350, doi: 10.1109/CCECE.2006.277525.

[7] E. Moser, M. K. Moran, E. Hillen, D. Li and Z. Wu, "Automatic modulation classification via instantaneous features," 2015 National Aerospace and Electronics Conference (NAECON), 2015, pp. 218-223, doi: 10.1109/NAECON.2015.7443070.

[8] L. Y. Uys, M. Gouws, J. J. Strydom and A. S. J. Helberg, "The performance of feature-based classification of digital modulations under varying SNR and fading channel conditions," 2017 IEEE AFRICON, 2017, pp. 198-203, doi: 10.1109/AFRCON.2017.8095481.

[9] Al-Nuaimi, D., Hashim, I., Zainal Abidin, I., Salman, L. and Mat Isa, N., 2019. Performance of Feature-Based Techniques for Automatic Digital Modulation Recognition and Classification—A Review. *Electronics*, 8(12), p.1407.

[10] Han, H., Ren, Z., Li, L. and Zhu, Z., 2021. Automatic Modulation Classification Based on Deep Feature Fusion for High Noise Level and Large Dynamic Input. *Sensors*, 21(6), p.2117.

[11] Han Gang, Li Jiandong and Lu Donghua, "Study of modulation recognition based on HOCs and SVM," 2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514), 2004, pp. 898-902 Vol.2, doi: 10.1109/VETECS.2004.1388960.

[12] B. Kim, J. Kim, H. Chae, D. Yoon and J. W. Choi, "Deep neural network-based automatic modulation classification technique," 2016 International Conference on Information and Communication Technology Convergence (ICTC), 2016, pp. 579-582, doi: 10.1109/ICTC.2016.7763537.

10. Appendix A ; Non-Technical Report

A. *Social Impact*

The Digital Divide is a disparity between regions which have modern communication system and areas which do not. South Africa is a developing country, and it has a large Digital divide. The Digital Divide is seen by the availability of communication systems in between the rural and urban areas.
South Africa is a country with large inequalities and the Digital Divide expands the inequalities. Rural areas have little to no access to fibre internet and weak cellular communication, whereas urban areas have fibre in every home

The Digital Divide was greatly expanded by the pandemic. An example of this situation is when government schools could not have classes online as the children were did not have adequate equipment or they lived in areas with poor networks, whereas private schools continued online.

The social impact of the Digital Divide is that the people with no access to the infrastructure have a disadvantage. The Fourth Industrial revolution ensured digitization of processes. This includes schooling and work. However, people who cannot overcome the disadvantage of the Digital Divide will have a lack of those crucial skills, thereby increasing the inequality.

Remote areas in which there are no cellular towers or results in a poor signal/network. This is due to the data received by the phone being corrupted because of the distance and the cellphone cannot recognize the data and deems it unreadable.

Automatic Modulation and Coding is a method introduced to solve the distance problem. It allows communication over a greater distance by altering communication parameters based on the distance. This allows for cellphone towers to reach a greater distance.

The receiver is required to know of the change in the communication parameters and thereby adapt to it. This introduces the concept of Automatic Modulation Recognition.

Automatic Modulation Recognition applies Machine Learning at the receiver to determine what type of communication parameters are being used. Automatic Modulation Recognition is used at the cellphone receiver in conjunction with Automatic Modulation and Coding at the cell tower to increase the cutoff distance.

Constructing cellular towers and laying out internet cables is expensive, therefore, the Digital Divide exists. Companies aim to maximize their reach to users, while remaining within a budget. The project allows companies to reach a larger audience, but also doing so in an economically viable manner. The project expands the capabilities of the current infrastructure, allowing it to reach a greater distance. It allows the current cellular towers to expand to further regions, thereby reducing the Digital Divide. This allows companies to maximize the “triple bottom line” in the balance sheet.

B. *Economical Impact*

Big Data has revolutionized the way businesses operate. Businesses are able to provide their customers with tailored services due to the large amount of information provided by big data. However, businesses do not house the infrastructure to compute the big data analytics. Companies like AWS provide a Cloud Service, which allows other businesses to use the computational power housed at the AWS data centers. The data is sent from the company to the AWS servers. Therefore, fast internet speeds are required.

To accommodate the growing trend in Cloud Computing, fast internet speeds are required from telecommunication companies. Automatic Modulation Recognition allows companies to supply the market-demand for fast internet speeds.

The shortage of chips may result in an increase in price, The chips are a bottleneck in production and may be economically damaging. Therefore, special contracts should be made with chip manufacturers in advance to avoid a drop in supply and radical price increases.

C. Environmental

AMR allow for the removal of excess hardware components which results in less waste during the end-of-life process. Removing excess hardware also reduces the power consumption and carbon footprint.

Chip manufacturers are changing from Silicon to Gallium, which is proven to be more energy efficient. Therefore, for long-term sustainability, Gallium chips should be considered

The first application of Automatic Modulation Recognition was in military warfare. The system does not require Automatic Modulation coding. A radio signal is sent between a WI-FI route and a phone. The system can be used to intercept communications between the router and the cellphone. The system is able to automatically detect the communication parameters with which the cellphone is communicating to the router. The system can then either transmit a signal to corrupt that signal , so the user does not receive a signal or intercept the signal and listen to the conversation and read the message.

The system is now being introduced into civilian telecommunications, where the social, economical and environmental benefits are seen.

.

11. Appendix B: Supervisor Meeting Minutes

Meeting 1: 02/08/2021

Talking Points

- Modulation Schemes
  QAM, PSK explained with fading and AWGN
- Features
  Different Features were described.
- Machine Learning
  Different Machine Learning techniques were described.
- Results
  Possible results were discussed
- Timeline

Action Items

- Input Data

Meeting 2: 10/08/2021

Talking Points

- Input
- Noise
  The Channel noise was explained
- Machine Learning Parameters
  Cost Functions described
- Channel Parameters
  Fading and Pulse Shaping explained

Action Items

- Input + Channel

Meeting 3: 16/08/2021

Talking Points

- MIMO
  MIMO systems described
- Feature Extraction
  Extraction of features from signal.
- Complexity
  Complexity of Algorithms explained

Action Items

- Extract Features

Meeting 4: 23/08/2021

Talking Points

- Results
  Possible Results Discussed
- Train Network
  Get results from system
- Sustainability
  Impact on society
- Fading Channel
  Fading Channel Explained

Action Items

- Train Network

Meeting 5: 23/08/2021

Talking Points

- Report
  Report key points stated
- Exit Levels Outcomes
  ELOs stated and explained
- Non-Technical Report
  Non-Technical Report explained

Meeting 6: 30/08/2021

Talking Points

- Report
  Report key points stated
- Exit Levels Outcomes
  ELOs stated and explained
- Non-Technical Report
  Non-Technical Report explained

12. APPENDIX C: MATLAB CODE

A. *Main.m*

```
clc
clear all

SNR=10;
NoFrames=1500;
frameSize=50;

%<--Generate and Modulate Signal------->

[psk, qam] = getdata(SNR,NoFrames,
frameSize);
%[psk, qam] =
get_rayleigh_data(SNR,NoFrames,
frameSize);
[pskt,qamt] = getdata(SNR,NoFrames,
frameSize);

%<-----Extract Features from Data---->

[features,label] = getFeatures(pskt,
qamt,NoFrames);
[featurest,labelt] =
getFeatures(pskt,qamt,NoFrames);

[classes, ia, labelnum]=unique(label);


%Thereafter, the Classification Learner
app is used to create the Neural.
%The Classification Learner App is a GUI
which allows you to create and
%test machine learning algorithms. It is
opened by typing
%classificationLearner in the command
window


%Network and the SVM using features as
an input and label as the target.
%The Algorithms are then tested using
the featurest dataset

%The SNR is manually changed after each
iteration to create a new dataset




function [psk,pskn,qam,qamn] =
getdata(SNR, NoFrames, frameSize)
%<--------Initialise Variables-------->

psk = [];
pskn=[];
qam = [];
qamn=[];


%<-------2 PSK DATA-------------->
M = 2;
data = randi([0 M-
1],NoFrames,frameSize);
txSig = pskmod(data,M,pi/M);
rpskSig = awgn(txSig,SNR);
noise = rpskSig - txSig;
pskn = [pskn; noise];
psk = [psk; rpskSig];


%<-----------4 PSK DATA---------------->
M = 4;
data = randi([0 M-
1],NoFrames,frameSize);
txSig = pskmod(data,M,pi/M);
rpskSig = awgn(txSig,SNR);
noise = rpskSig - txSig;
pskn = [pskn; noise];
psk = [psk; rpskSig];


%<%<--------16 QAM DATA---------->
M = 16;
data = randi([0 M-
1],NoFrames,frameSize);
y = qammod(data,M,
'UnitAveragePower',true);
rqamSig = awgn(y,SNR);
n = rqamSig - y;
qam = [qam; rqamSig];
qamn = [qamn; n];


 %<%<-----64 QAM DATA------------->
M = 64;
data = randi([0 M-
1],NoFrames,frameSize);
y = qammod(data,M,
'UnitAveragePower',true);
rqamSig = awgn(y,SNR);
n = rqamSig - y;
qam = [qam; rqamSig];
qamn = [qamn; n];


end



function [features,label] =
getFeatures(psk,qam,NoFrames)
```

```
%<--------Initialise Variables-------->
label=[];

%<--------Get Cumulants for PSK-------->
pskfeatures = get_cumulants(psk);

%<--------Label the PSK Features--------
>
pskfeatures =
pskfeatures(1:size(psk,1),1:9);
strq=repmat('oobpsk',NoFrames,1);
label = [label;strq];
strq=repmat('oo4psk',NoFrames,1);
label = [label;strq];

%<--------Get Cumulants for PSK-------->
qamfeatures = get_cumulants(qam);

%<--------Label the PSK Features--------
>
qamfeatures =
qamfeatures(1:size(qam,1),1:9);
strq=repmat('o16qam',NoFrames,1);
label = [label;strq];
strq=repmat('064qam',NoFrames,1);
label = [label;strq];

label = cellstr(label);
features = [pskfeatures;qamfeatures];

end

function [cumulants] =
get_cumulants(signalData)
```

```
%<--------Initialise Variables-------->
shape = size(signalData);
H = shape(1);
L = shape(2);
cumulants = zeros(H, L);

%<--------Calculate Cumulants-------->
for row = 1:H
    M20 = sum(signalData(row,:).^2)/L;
    M21 =
sum(abs(signalData(row,:)).^2)/L;
    M22 =
sum(conj((signalData(row,:))).^2)/L;
    M40 = sum(signalData(row,:).^4)/L;
    M41 =
sum(abs(signalData(row,:)).^2.*signalDat
a(row,:).^2)/L;
    M42 =
sum(abs(signalData(row,:)).^4)/L;
    M43 =
sum(abs(signalData(row,:)).^2.*conj((sig
nalData(row,:))).^2)/L;
    M60 = sum(signalData(row,:).^6)/L;
    M61 =
sum(abs(signalData(row,:)).^2.*signalDat
a(row,:).^4)/L;
    M62 =
sum(abs(signalData(row,:)).^4.*signalDat
a(row,:).^2)/L;
    M63 =
sum(abs(signalData(row,:)).^6)/L;
    C20 = M20;
    C21 = M21;
    C40 = M40 - 3*M20^2;
    C41 = M41 - 3*M20*M21;
    C42 = M42 - abs(M20)^2 - 2*M21^2;
    C60 = M60 - 15*M20*M40 + 3*M20^3;
    C61 = M61 - 5*M21*M40 - 10*M20*M41 +
30*M20^2*M21;
    C62 = M62 - 6*M20*M42 - 8*M21*M41 -
M22*M40 + 6*M20^2*M22 + 24*M21^2*M20;
    C63 = M63 -9*M21*M42 + 12*M21^3 -
3*M20*M43 - 3*M22*M41 + 18*M20*M21*M22;
    C21_modify = C21 -
var(signalData(row,:));
    C20_norm = C20/(C21_modify^2);
    C21_norm = C21/(C21_modify^2);
    C40_norm = C40/(C21_modify^2);
    C41_norm = C41/(C21_modify^2);
    C42_norm = C42/(C21_modify^2);
    C60_norm = C60/(C21_modify^2);
    C61_norm = C61/(C21_modify^2);
    C62_norm = C62/(C21_modify^2);
    C63_norm = C63/(C21_modify^2);
    cumulants(row, 1) = abs(C20_norm);
    cumulants(row, 2) = abs(C21_norm);
    cumulants(row, 3) = abs(C40_norm);
    cumulants(row, 4) = abs(C41_norm);
    cumulants(row, 5) = abs(C42_norm);
    cumulants(row, 6) = abs(C60_norm);
    cumulants(row, 7) = abs(C61_norm);
    cumulants(row, 8) = abs(C62_norm);
```

```
    cumulants(row, 9) = abs(C63_norm);

    cumulants(row, 1) = abs(C20);
    cumulants(row, 2) = abs(C21);
    cumulants(row, 3) = abs(C40);
    cumulants(row, 4) = abs(C41);
    cumulants(row, 5) = abs(C42);
    cumulants(row, 6) = abs(C60);
    cumulants(row, 7) = abs(C61);
    cumulants(row, 8) = abs(C62);
    cumulants(row, 9) = abs(C63);
end



function [psk,pskn,qam,qamn] =
get_rayleigh_data(SNR, NoFrames,
frameSize)
```

```
%<--------Initialise Variables-------->

psk = [];
pskn=[];
qam = [];
qamn=[];

%<--------Initialise Channel-------->

rchan =
comm.RayleighChannel('SampleRate',1000,
...
    'PathDelays',[0 1e-3 2e-
3],'AveragePathGains',[0 -3 -6], ...
    'MaximumDopplerShift',0, ...
    'RandomStream','mt19937ar with
seed','Seed',73);


%<--------------2 PSK DATA------------->
M = 2;
data = randi([0 M-
1],NoFrames,frameSize);
txSig = pskmod(data,M);
for i = 1:NoFrames
t = txSig(i,:);
t = t';
t = rchan(t);
t = t';
txSig(i,:) = t;
end
rpskSig = awgn(txSig,SNR);
%scatterplot(rpskSig(1,:))
noise = rpskSig - txSig;
pskn = [pskn; noise];
psk = [psk; rpskSig];


%<--------------4 PSK DATA------------->
M = 4;
data = randi([0 M-
1],NoFrames,frameSize);
txSig = pskmod(data,M);
for i = 1:NoFrames
t = txSig(i,:);
t = t';
t = rchan(t);
t = t';
txSig(i,:) = t;
end
rpskSig = awgn(txSig,SNR);
%scatterplot(rpskSig(1,:))
noise = rpskSig - txSig;
pskn = [pskn; noise];
psk = [psk; rpskSig];



%<%<------------16 QAM DATA------->
M = 16;
data = randi([0 M-
1],NoFrames,frameSize);
```

```
txSig =
qammod(data,M,'UnitAveragePower',true);
for i = 1:NoFrames
t = txSig(i,:);
t = t';
t = rchan(t);
t = t';
txSig(i,:) = t;
end
rqamSig = awgn(txSig,SNR);
n = rqamSig - txSig;
qam = [qam; rqamSig];
qamn = [qamn; n];


%<%<------------64 QAM DATA-------->
M = 64;
data = randi([0 M-
1],NoFrames,frameSize);
txSig =
qammod(data,M,'UnitAveragePower',true);
for i = 1:NoFrames
t = txSig(i,:);
t = t';
t = rchan(t);
t = t';
txSig(i,:) = t;
end
rqamSig = awgn(txSig,SNR);
%scatterplot(rqamSig(1,:))
n = rqamSig - txSig;
qam = [qam; rqamSig];
qamn = [qamn; n];


end
```