FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

MASTER'S THESIS

PREDICTIVE RESOURCE ALLOCATION FOR URLLC USING EMPIRICAL MODE DECOMPOSITION

Author	Chandu Jayawardhana
Supervisor	Prof. Nandana Rajatheva
Second Examiner	Dr. Nurul Huda Mahmood
Technical Advisor	Thushan Sivalingam

November 2022

Jayawardhana C. (2022) Predictive Resource Allocation for URLLC using Empirical Mode Decomposition. University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Wireless Communications Engineering, 39 p.

ABSTRACT

Empirical mode decomposition (EMD) based hybrid prediction methods can be an efficient way to allocate resources for ultra reliable low latency communication (URLLC). In this thesis, we have considered efficient resource allocation for the downlink channel at the presence of several interferers. Initially, we have generated desired signal that we need to transmit in downlink and total interference signal that will affect the desired signal transmission. Then, we used EMD to decompose the total interference signal power into intrinsic mode functions (IMFs) and residual. Due to the properties of EMD, decomposed IMFs become less random as IMF number increases. As a result of that property, prediction model training process become less complex and prediction accuracy also increases as randomness of signal decreases. Long short term memory (LSTM) deep neural network method and auto regressive integrated moving average (ARIMA) time series method are deployed to predict future interference power values based on historical values. For each decomposed component (IMFs and residual), two prediction models have been trained using LSTM and ARIMA methods. Finally, predicted components of IMFs and residual are added together to form total predicted interference power.

According to the predicted interference power, resources are allocated for downlink transmission of the signal and evaluated it with the baseline estimation techniques. The research demonstrates that the suggested method achieves near optimal resource allocation for URLLC.

Keywords: Auto regressive integrated moving average, block length theory, empirical mode decomposition, intrinsic mode functions, long short term memory, residual, ultra reliable low latency communication.

TABLE OF CONTENTS

FOREWORD

This thesis focused on predictive resource allocation for ultra reliable low latency communication using empirical mode decomposition. This work was supported by the project AL/ML for Beamforming in 6G (grant no. 24303959) and the Academy of Finland 6Genesis Flagship (grant no. 346208). This is a perfect opportunity for me to convey my appreciation to my thesis supervisors Prof. Nandana Rajatheva , Dr. Nurul Huda Mahmood, and technical advisor Thushan Sivalingam. The aforementioned team guided me throughout the masters thesis course to make my thesis a success. My supervisors dependably allowed me to select the thesis subject in my interest area and supported and guided me in the right direction to achieve the end goal. I appreciate all the professors at the University of Oulu for their immense support. Finally, I would like to thank my parents and colleagues for their encouragement and support throughout my research period and during the master's degree period. These achievements would not have been possible without you. Thank you.

Oulu, 2nd December, 2022

Chandu Jayawardhana

LIST OF ABBREVIATIONS AND SYMBOLS

Acronyms

ACF	Auto Correlation Function
AR	Augmented Reality
ARIMA	Auto Regressive Integrated Moving Average
AWGN	Additive White Gaussian Noise
BIC	Bayesian Information Criterion
BLER	Block Error Rate
BS	Base Station
CA	Carrier Aggregation
CSI	Channel State Information
DNN	Deep Neural Network
EMD	Empirical mode decomposition
E2E	End to End
FF	Forgetting Factor
IIR	Infinite Impulse Response
IMF	Intrinsic Mode Functions
INR	Interference to Noise Ratio
ITU	International Telecommunication Union
LSTM	Long Short Term Memory
LTE	Long Term Evolution
MA	Moving Average
MEC	Mobile Edge Computing
MIMO	Multiple Input Multiple Output
MSE	Mean Squared Error
NB-IoT	Narrow Band Internet Of Things
NN	Neural Networks
PACF	Partial Auto Correlation Function
PCA	Principal Component Analysis
QoS	Quality of Service
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SDN	Software Defined Network
SINR	Signal to Interference and Noise Ratio
SMA	Simple Moving Average
SNR	Signal to Noise Ratio
TTI	Transmission Time Interval
UE	User Equipment
URLLC	Ultra Reliable Low Latency Communication
VMD	Variational Mode Decomposition
VR	Virtual Reality
eMBB	enhanced Mobile Broadband
mMTC	massive Machine-Type Communication
1G	First Generation

4G	Fourth Generation
5G	Fifth Generation

Symbols

L	Addition of IMFs and residual
$\varepsilon_{achieved}$	Achieved block error probability
ε	Block error rate / Target outage
Y_t	Current value of ARIMA method
ϕ	Constants of AR method
c	Constant in AR and MA methods
θ	Constants of AR method
$V(\gamma)$	Channel dispersion
$\bar{\gamma}$	Desired signal SINR
$\hat{\gamma}$	Estimated SINR value
α	Forgetting factor of IIR filter
X_t	Input to the recurrent neural network
γ_{min}	Minumum INR value of interfering link
γ_{max}	Maximum INR value of interfering link
N_0	Normalized noise power
p	Number of previous samples to use in AR method
q	Number of previous samples to use in MA method
n	Number of time steps in neural networks
D	Number of bits
R	Number of resources
T	Number of interference samples considered for simulation
N	Number of samples in training data
M	Number of samples in validating data
N_0	Normalized noise power
d	Order of integration in ARIMA method
h_t	Output of the Recurrent neural network
$\hat{I}_{IIR}$	Predicted interference using moving average estimation
$\hat{I}_{Genie}$	Predicted interference using genie aided estimation
$\hat{I}$	Predicted interference power
$\hat{\gamma}$	Predicted SINR value
Q	Q-function
W	Scalar value in neural networks
$C(\gamma)$	Shannon capacity of AWGN under a finite block length regime
S	Signal power of the desired signal

Operators

$\lvert \cdot \rvert$	Absolute value
$\sum$	Summation operation

1 INTRODUCTION

Demand for higher data rates was the main driving factor over the past generations (first generation (1G) to fourth generation (4G)) of telecommunication. In the latter part of 4G, different applications and services emerged that require more diverse infrastructure resource support from wireless technology than the higher data rates [1]. Some applications are autonomous vehicles, drone-based deliveries, smart cities and factories, remote medical diagnosis and surgery, and artificial intelligence-based personalised assistants [2]. A telecommunication network must have features like massive connectivity, high reliability, low latency, and greater energy efficiency to facilitate the above applications. To address these diversified feature requirements, the international telecommunication union (ITU) has categorised fifth generation (5G) services into three main streams: ultra-reliable low latency communication (URLLC), massive machine-type communication (mMTC), and enhanced mobile broadband (eMBB) [3, 4].

Among the three streams mentioned above, the design of URLLC to simultaneously meet high-reliability and low-latency requirements is the most challenging task [5]. To improve reliability, additional resources must be devoted for signaling, re-transmission, redundancy, and parity checks, which leads to increase redundancy. Because many URLLC applications' throughput requirements are not stringent, when URLLC packet size is small, we can achieve high reliability without exceeding latency targets by utilizing resources in the frequency, antenna, and spatial domain [2].

The primary objective of this thesis is to provide an up-to-date summary of URLLC developments and advancements by emphasizing potential obstacles and proposing feasible solutions to overcome those challenges. First, we briefly describe the three service classes of 5G, and then we discuss the issues and enabling technological solutions. This remaining chapters of this thesis are structured as follows. Chapter 2 discusses the theories and methods we are going to use to solve the identified issues. Chapter 3 represents the system model and we propose the EMD based prediction and resource allocation scheme. Simulation results and performance analysis are given in Chapter 4. Conclusions are presented in Chapter 5.

The eMBB service category has been defined to address the conventional requirements of higher data rates and high bandwidth. This is useful for applications such as high-resolution video streaming, virtual reality (VR), and augmented reality (AR). The following physical layer technologies were introduced to address the above application requirements: higher-order modulation transmission schemes, carrier aggregation (CA), cell densification, massive multiple input multiple output (massive MIMO) transmission [6], millimeter-wave communication [7], and use of unexplored spectrum.

The mMTC service category has been defined to provide access to a substantial number of machine-type devices. This is useful for applications such as sensing, tagging, metering, and monitoring that require high connection density and better energy efficiency [8]. Throughout the years, a number of technology solutions have emerged to satisfy the aforementioned needs, including narrowband internet of things (NB-IoT) in a licensed band and SigFox and LoRa in an unlicensed band [9]. In addition to supporting the above mentioned requirements, it facilitates benefits like reduced power consumption, low operation cost, and enhanced coverage. However, the performance of the mMTC service degrades when the number of devices outweighs the resources. To address this issue, an aggressive connection strategy violating the orthogonal transmission principle is

required [2]. More user accommodation with limited number of resources can be realized by using non-orthogonal spreading sequences and user-specific interleaving [8, 10, 11].

The URLLC service category has been defined to assist mission-critical applications that require uninterrupted, reliable, and fast data exchange. It is useful for factory automation, process automation, smart grids, intelligent transport systems, and professional audio [9]. Although latency has improved from third generation (3G) to 4G, long term evolution (LTE) end-to-end (E2E) latency is still in the range of $30 - 100$ ms. Most of the URLLC services require E2E latency to be in the range of 1 ms. The targeted quality of service (QoS) requirement in URLLC is $1 - 10^{-5}$ [12]. To address the latency requirement, fundamental changes in wireless links and backbone networks are suggested. In the backbone link, software-defined networking (SDN) and virtual network slicing can be used to establish a private connection to the URLLC service [8]. Thus, the backbone link latency can be decreased. In wireless networks, overheads must be minimized and the transmission process must be optimized to decrease latency.

When we try to achieve the expected reliability and latency requirements of URLLC, there are some key challenges associated with it. Those are: performance degradation of the other two services (eMBB, mMTC) when prioritizing URLLC, unable to achieve the expected 1 ms E2E delay due to routing, and propagation delays in backhauls and core networks. To address the above challenges, solutions such as mobile edge computing (MEC), SDN, network slicing, multi-connectivity, and anticipatory networking have been proposed [13]. However, some of the proposed solutions also have limitations at the implementation stage. For example, MEC implementation is difficult due to complex optimization due to the non-convexity of the problem [14], and task offloading is difficult due to high user mobility and high overhead on exchanging information. Deep learning frameworks have been widely used to solve the above-mentioned limitations of the proposed solutions. The deep learning framework can be categorised into four sub-categories: federated learning, deep transfer learning, lifelong deep learning, and distributed deep learning [13].

Therefore, simultaneously achieving low latency and high reliability is crucial for future improvements and enhancements of URLLC not only in 5G but also in future generations such as sixth generation (6G). Reducing re-transmissions of URLLC packets is one way to achieve the high reliability while minimizing latency [15]. Interference signals will lead to data packet collisions. To achieve the reliability targets packets must be re-transmitted at the expense of ensuring low latency. If we can predict the interference signal that will affect the desired transmission, we will be able to reduce the number of re-transmissions by efficiently allocating the resources that can withstand the interference while meeting the reliability targets.

In our study, we addressed the URLLC's latency and reliability issues and proposed an effective resource allocation mechanism in the downlink.

1.1 Thesis Contribution

This thesis is conducted on the topic of predictive resource allocation for URLLC using EMD, to efficiently allocate resources for URLLC services in downlink channel. The system model consists of N number of interfering links with one desired link. Base station (BS) is trying to allocate downlink resources as efficiently as possible to minimize the effect of actual interference by predicting possible interference values and pre-allocating resources. Our goal is to implement an algorithm for above predictive resource allocation purpose to ensure both reliability and latency targets of the URLLC.

1.2 Thesis Outline

The remainder of the thesis is structured as follows:

- **Chapter 2:** All the necessary concepts, associated theories, summaries of prior works, and study proposals are included.
- **Chapter 3:** In depth analysis of the study is provided. It includes a system model, a problem statement, a solution approach, and performance evaluation criteria.
- **Chapter 4:** The results of the study are discussed in this chapter. addressing the problem using simulation tools and benchmarking of results are presented.
- **Chapter 5:** In this chapter, we summarise the contribution of our study and suggest potential areas for further investigation.

2 BACKGROUND

This chapter depicts the principles associated with our study and assessment about using empirical mode decomposition (EMD) to predict URLLC interference and resource allocation. Existing 5G URLLC enabling technologies separately focus on achieving low latency and high reliability. The independently created reliability and latency-enabling technologies can collectively meet the 5G standards. Also, note that beyond 5G requirements have a stricter cap than 5G. Therefore, under efficient and scalable resource allocation beyond 5G communication, the current 5G latency and reliability criteria are unacceptable. For example, 5G achieves reliability requirements by over-provisioning resources, which otherwise could have been allocated for some other service, such as eMBB or mMTC. To ensure the efficient allocation of resources for URLLC traffic, it is crucial to anticipate its interference beforehand. Then we can preallocate the required resources for URLLC traffic and reduce the effect of the interference. Prediction algorithms must be implemented to identify the URLLC interference in advance.

Due to the complexity and uncertainty of the URLLC interference, decomposition-based combination prediction models have been selected. These decomposition methods include EMD, principal component analysis (PCA), dynamic tensor completion, spectral analysis, wavelet transform methods, variational mode decomposition (VMD) [16]. These approaches deconstruct the original interference pattern into multiple distinct components. Then, the relevant prediction algorithms for each component are selected. Finally, predicted component values are combined to obtain the total prediction.

In wavelet transform, since it is not straightforward to select the right wavelet basis function and decomposition level, its adaptability is low. In VMD, we need to predefine the number of decomposition levels, but in practice its hard to define ideal number of decomposition levels beforehand. Although PCA is capable of preserving the original data's information content to the greatest extent, it cannot be used with nonlinear interference signals since it is a linear dimension reduction technique. Therefore, EMD is the most suitable method for URLLC traffic signal decomposition due to its ability to process non-linear and non-stationary data. Modes obtained by the EMD have physical meaning and importance, while modes obtained form other decomposition methods are mathematically important but difficult to relate to the real world.

2.1 Empirical Mode Decomposition

EMD decomposes given data into intrinsic mode functions (IMF) using the heuristic decomposition method. Because these functions are not set analytically but determined by an analyzed sequence alone, the term "empirical" refers to that property. Therefore, essential functions are derived adaptively directly from the input sequence. The IMF's derived from EMD needs to satisfy the below conditions [17].

- The total number of minima and maxima needs to be equal or differ at most by one with the number of zero crossings.
- At any given time, the IMF mean value derived from envelopes formed by local maxima and local minima must equal zero.

As a result of this decomposition method, we can obtain frequency-ordered IMF components. Initial few IMFs contain high-frequency oscillations, and the rest have low-frequency components, where oscillation frequency reduces as the IMF number increases, as shown in Figure 2.1. Finally, only one component with a mean non-zero element remains at the end, called the residue. Due to the above property of EMD, decomposed signals become less random as the IMF number increases. Developing a model to predict a less random (linear) signal over a random (non-linear) signal is much easier and more accurate. Therefore, it is advantageous to use the decomposed signals for prediction. To clearly understand and visualize the step-by-step procedure of EMD, we show the complete decomposition of a particular signal in Figure 2.1. Further, the mathematical background and the algorithms are discussed below.

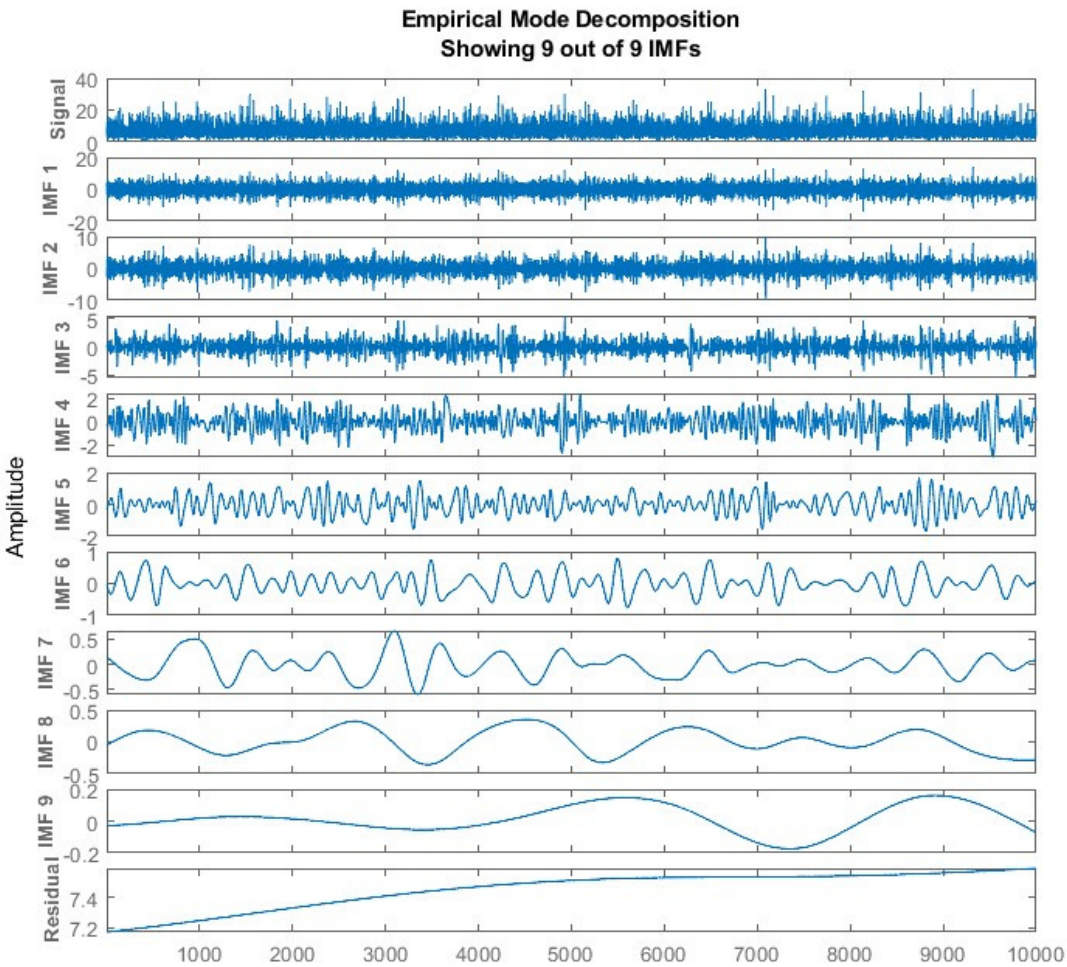

Figure 2.1. Empirical mode decomposition of a signal.

The first step of the EMD algorithm is to find all the local extremas of the signal (local maxima and minima). Next, we fit two envelopes, one with the selected maximas and another with the selected minimas. Typically, cubic spline functions are used to fit the envelopes, but we can use other functions that cover all the data in between them. The

mean envelope is then calculated from the upper and lower envelopes at each point in time, as given by

$$E_{mean}(t) = \frac{E_{up}(t) + E_{low}(t)}{2}, \tag{1}$$

where $E_{up}(t)$ is the upper envelope, $E_{low}(t)$ is the lower envelope, and $E_{mean}(t)$ is the mean envelope. The above three envelopes are shown in the Figure 2.2.

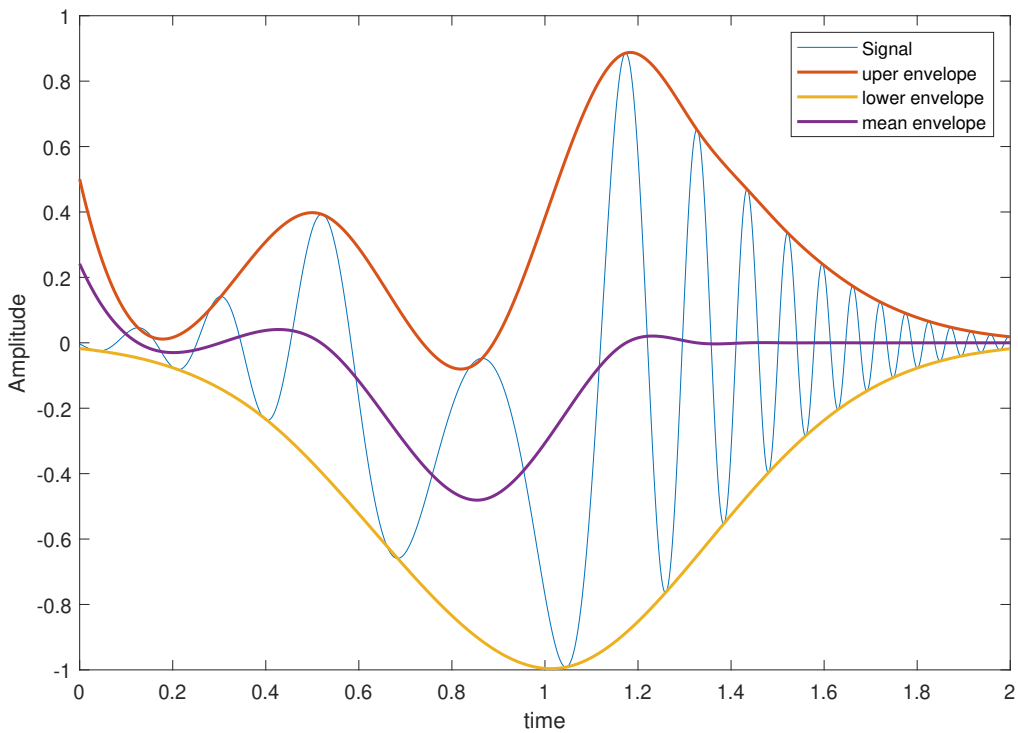

Figure 2.2. Maxima envelope, minima envelope, and mean envelope of a signal.

Finally, the residual is calculated by subtracting the previously calculated mean from the original signal as

$$res(t) = f(t) - E_{mean}(t), \tag{2}$$

where the original signal is $f(t)$ and the obtained residual value is $res(t)$.

Next, we must determine if the residual obtained from the preceding equation has a mean of zero. If it has a mean of zero, we consider it to be our initial IMF. To determine this, a stopping criterion is applied. In the initial version of the EMD, the stopping criterion is essentially a standard deviation that takes into account the squared difference of the original signal $f(t)$ and the newly obtained residual normalised by the squared signal. The values are summed over all time steps t, and the resulting single outcome is compared to a threshold value ϵ as follows

$$\sum_{t=1}^{\infty} \frac{(res(t) - f(t))^2}{f(t)^2} < \epsilon. \tag{3}$$

Usually, the first residual we obtain does not possess the characteristics of an IMF and the stopping criteria are not satisfied. This is because extrema fitting often produces

over and undershoots. Before we can call this first residual an IMF, we must eliminate newly formed extremes. If the stopping criterion is not met, we repeat the process with the residual $res(t)$ until the stopping criterion is met, as shown in below

$$E_{mean_2}(t) = \frac{E_{up}(res(t)) + E_{low}(res(t))}{2}, \tag{4}$$

$$res_2(t) = res(t) - E_{mean_2}(t), \tag{5}$$

$$\sum_{t=1}^{\infty} \frac{(res_2(t) - res(t))^2}{res(t)^2} < \epsilon. \tag{6}$$

During the above stopping criterion check (6), if it satisfies the conditions, we update the residual as our first IMF

$$IMF_1(t) = res_2(t), \tag{7}$$

Where $res_2(t)$ is the residual value which satisfied the stopping criterion check. Then, we subtract $IMF_1(t)$ from the original signal to obtain an updated version of signal $f(t)$ for use of finding the remaining IMFs as

$$f_{update1}(t) = f(t) - IMF_1(t). \tag{8}$$

Once $IMF_2(t)$ is determined, it needs to be subtracted from the previously updated signal $f_{update1}(t)$ to obtain a new updated version of the signal $f_{update2}(t)$ and so on. Typically, we stop the above process when we obtained a monotonic function as a residual and we are unable to extract extremas anymore. Thus, at the end of the process, only one residual remains. Finally, the original signal $f(t)$ can be recreated by adding all IMFs and the residual as shown by

$$f(t) = \sum_{i=1}^{L-1} IMF_i(t) + res(t), \tag{9}$$

where L is the number of decomposed components.

2.2 Time Series Forecasting

Time is the most crucial factor upon which the majority of critical parameters depend. Since people are more interested in predicting the future, various prediction mechanisms have been developed over time. "Time Series Data" refers to the data that has been collected at evenly spaced time intervals. A time series is any parameter that is ordered sequentially over time at consistent intervals, such as hours, days, weeks, or months. "Time Series Forecasting Techniques" refers to the methods we use to predict future values based on the preceding time series data. There are numerous statistical time series forecasting methods and a few of those are listed below [18].

- Simple Moving Average (SMA).
- Exponential Smoothing.

- Autoregressive Integration Moving Average (ARIMA).
- Neural Networks (NN).

2.3 Auto Regressive Integrated Moving Average (ARIMA)

Due to its ability to work with non-stationary data sequences, ARIMA is a widely used and popular time series forecasting method. The ARIMA model applies an equation to time series data for forecasting purposes. The fitted model can be one of three varieties [19].

- Autoregressive model (AR).
- Moving Average Model (MA).
- mixed of AR and MA models.

The AR model linearly predicts the current value of the variable based on its previous values as far back as p periods, as demonstrated by

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t, \tag{10}$$

where Y_t is the current value and (Y_t,Y_{t-1}...,Y_{t-p}) are previous values. Constants ϕ_1 to ϕ_p and c must be estimated in this instance and denoted by **AR(p)**.

The moving average model linearly predicts the current value Y_t by calculating the weighted average of past errors heading back q periods as demonstrated by

$$Y_t = c + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} ..... - \theta_q e_{t-q}, \tag{11}$$

where (e_t,e_{t-1}...,e_{t-q}) are past errors. Constants θ_1 to θ_q and c must be estimated in this instance and denoted by **MA(q)** . By combining the AR and MA models described above, the ARMA model can be created as

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} + \theta_2 e_{t-2} + ... + \theta_q e_{t-q}. \tag{12}$$

The data used in ARIMA modelling must be stationary. In other words, an upward or downward trend should not exist in the time series data, and the variable's value should fluctuate around a constant mean. Since most practical data is non-stationary and exhibits a trend, the trend must be eliminated before its use in this model. To eliminate the trend and produce a stationary data set, the ARIMA model subtracts the data that varies from one period to the next, as shown by

$$\Delta Y_t = Y_t - Y_{t-1}, \tag{13}$$

where ΔY_t is the first difference. Typically, the first difference in non-stationary data is stationary. If the first difference is also non-stationary, however, the second difference may be utilised. Order of the integration refers to the number of times the difference must be taken to arrive at a stationary data set; I in ARIMA, parameter d in ARIMA (p,d,q) model denotes this. To obtain accurate forecasts using the ARIMA model, the p, d, q parameters of the dataset must be optimised. The process of ARIMA modeling includes four basic steps

- **Identification** First, determine whether the data is stationary; if not, calculate the difference until the remaining data becomes stationary. Then, determine the values for p and q. There is no exact method for determining these values. We must instead rely on our judgement and certain rules. The value of d is typically one, but it can vary depending on the data set.
- **Estimation** To estimate the coefficient values θ and ϕ, various types of models may be employed. (e.g SPSS software)
- **Diagnostics** The residuals, which represent an estimate of the error terms, must be completely random. In ARIMA modelling, the autocorrelation function (ACF) and partial autocorrelation function (PACF) are utilised to assess randomness [20]. Since multiple values of p and q may result in the residual having a completely random pattern, the bayesian information criterion (BIC) is utilised to discover the optimal combination. As the preferred model, the p and q value pair with the lowest BIC will be chosen.
- **Forecasting** Use formulated model equation to calculate forecasts.

2.4 Neural Networks

A neural network can be defined as a differentiable mathematical function that maps one type of variable to another. In classification problems, vectors are converted to vectors. In regression problems, vectors are transformed into scalars. Recurrent networks simply transform sequences into a variety of outputs. Consequently, we can observe the following application-specific architectures:

- **Vector to sequence model** Here, a vector is used to generate a sequence of the specified length. This method is used for image captioning. It takes a vector representation of an image as input and generates a sequence of words describing or captioning that image as output.
- **Sequence to vector model** It accepts a sequence as input and returns a vector of fixed length. This method is frequently employed in sentiment analysis, where it takes as input a sequence of words or phrases and returns a vector describing the sentiment (mostly two-dimensional vector stating whether the sentiment is positive or negative).
- **Sequence to sequence model** In this case, both inputs and outputs are sequences. This method is frequently employed by text prediction platforms to predict the next word in an input sequence. In this method, the length of the sequence is crucial. Theoretically, the length of the sequences can be infinite, but there are some practical issues, as illustrated by the following Section 2.4.1.

2.4.1 Vanishing and exploding gradient problem in neural networks

Consider a simple recurrent network with no hidden units other than the repetition of some scalar $X^{(0)}$. After n time steps its value would be $X^{(n)}$. In discrete dynamic systems

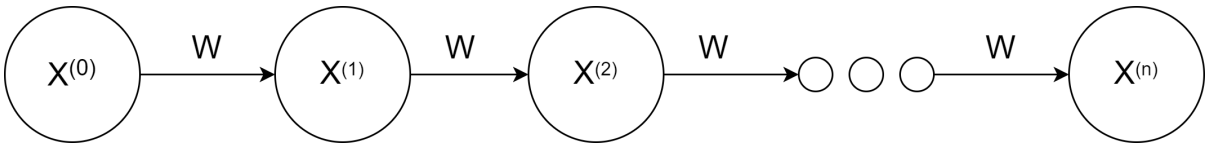

Figure 2.3. Simple recurrent network with no hidden units.

we can write $X^{(n)}$ as

$$X^{(n)} = W^n X^{(0)}, \tag{14}$$

where W is a scalar. To determine the scalar W, we must employ back propagation through the time. At large values of n, the value of $X(n)$ depends on the value of W. If the value is greater than 1, $X^{(n)}$ tends to explode, and if it is less than 1, $X(n)$ would vanish.

$$W^n X^{(0)} \rightarrow \begin{cases} \infty; & W > 1 \\ 0; & W < 1 \end{cases} \tag{15}$$

The same will also be applied for its gradients as

$$\frac{\partial W^n X^{(0)}}{\partial W} \rightarrow \begin{cases} \infty; & W > 1 \\ 0; & W < 1 \end{cases} \tag{16}$$

These results can be generalized to matrices as well. For W entries with values greater than 1, the corresponding eigenvectors of $W^{(n)}$ will then explode. This signifies that the input values in the direction of the eigenvectors will explode to infinity and information will be lost. For values of W less than one, the opposite is observed: the corresponding eigenvectors approach 0 and the input components in the direction of these eigenvectors vanish, leading to a loss of input information. To address the problem of vanishing and exploding gradients in recurrent neural networks, researchers have developed the Long Short Term Memory method (LSTM).

2.4.2 Recurrent Neural Network (RNN)

RNN architecture was created for the first time in the 1980s [21]. There are three types of layers in this architecture: input layer, one or more hidden layers, and output layer. The RNN is composed of modules that are repeated and arranged in a chain structure. The significance of this module structure is that it can be used as memory cells to store previous data, which is essential for making predictions. The RNN is comprised of an input-accepting feedback loop. This feedback loop enables the effect of the previous state's outcome to influence the current state's outcome. This effect propagates throughout the chain due to its chain-like structure, resulting in successful learning.

Figure 2.4 depicts the simple RNN with single input, single output, and single recurrent hidden unit that has grown to a complete network. X_t represents the input at time t, whereas h_t represents the output. This RNN uses a backpropagation algorithm during the training process.

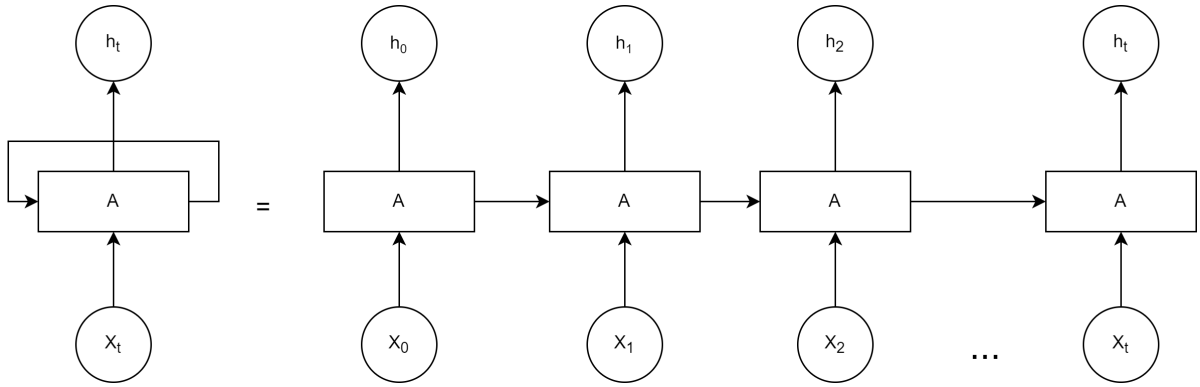

Figure 2.4. Sequential processing in RNN.

2.4.3 Long Short Term Memory (LSTM)

Hochreiter and Schmidhuber invented LSTM [21] to solve the limitation of conventional RNNs, such as vanishing or exploding gradient problems. LSTM is a specialised RNN capable of learning long-term dependencies and carrying forward learned information for a long period by supporting future decisions. As traditional RNNs, LSTM also has a chain structure. However, the repeating module of LSTM has a different structure compared to a traditional RNN.

In addition to the hidden state vector, every LSTM cell contains a cell state vector. Using an explicit gating mechanism, the subsequent LSTM may choose to read from, write to, or even reset the memory at each time step. There are three types of LSTM cell gates.

- **Input Gate** Control whether the memory cell is updated.
- **Forget Gate** Controls whether the memory cell is reset to zero.
- **Output Gate** Controls whether information of the current cell state is made visible.

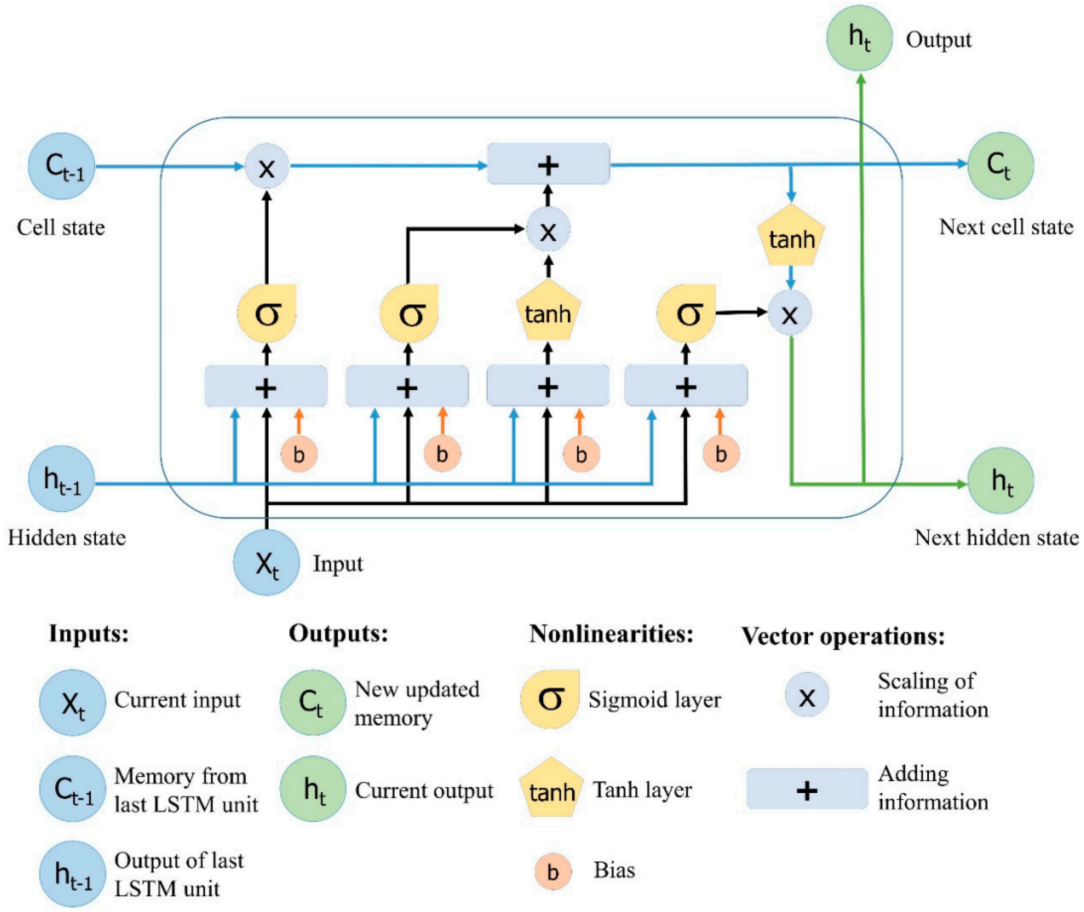

Figure 2.5. LSTM Structure [21].

All three type of gates described above and included in Figure 2.5 have sigmoid activation functions due to the smoothness of the sigmoid curve in the range of zero to one $[0, 1]$. Other than the above gates, there is additional vector associated with LSTM that alter the cell state. It possess the tanh activation function due to its capability to distribute the gradients with a zero-centered range for long sequences. It allows cell state information to flow for a longer time without vanishing or exploding. Each gate in an LSTM cell receives a current input and a hidden state as inputs. After concatenating the vectors, the sigmoid is applied. The aforementioned vector is then applied to the cell state to introduce candidate values.

3 URLLC INTERFERENCE PREDICTION AND RESOURCE ALLOCATION

3.1 System Model

In this study, the downlink of a wireless network is examined. The objective is to determine the optimal resource allocation for a URLLC service operating in the presence of N interferers, as shown in Figure 3.1. It is assumed that the desired channel has a signal-to-noise ratio (SNR) of $\bar{\gamma}$ and the interference-to-noise ratios (INR) for each interfering link are uniformly distributed within a given range $[\gamma_{min}, \gamma_{max}]$. The previously mentioned INR is also known as the SNR of the interference signal.

The desired link is assumed to be established with the base station which has the highest SNR. Therefore, it is assumed that the interfering link with the highest SNR value (γ_{max}) has a lower SNR value compared to the desired link ($\bar{\gamma} > \gamma_{max}$). In addition, it is assumed that there is no cooperation between transmitters, and the single antenna Rayleigh block fading method is employed.

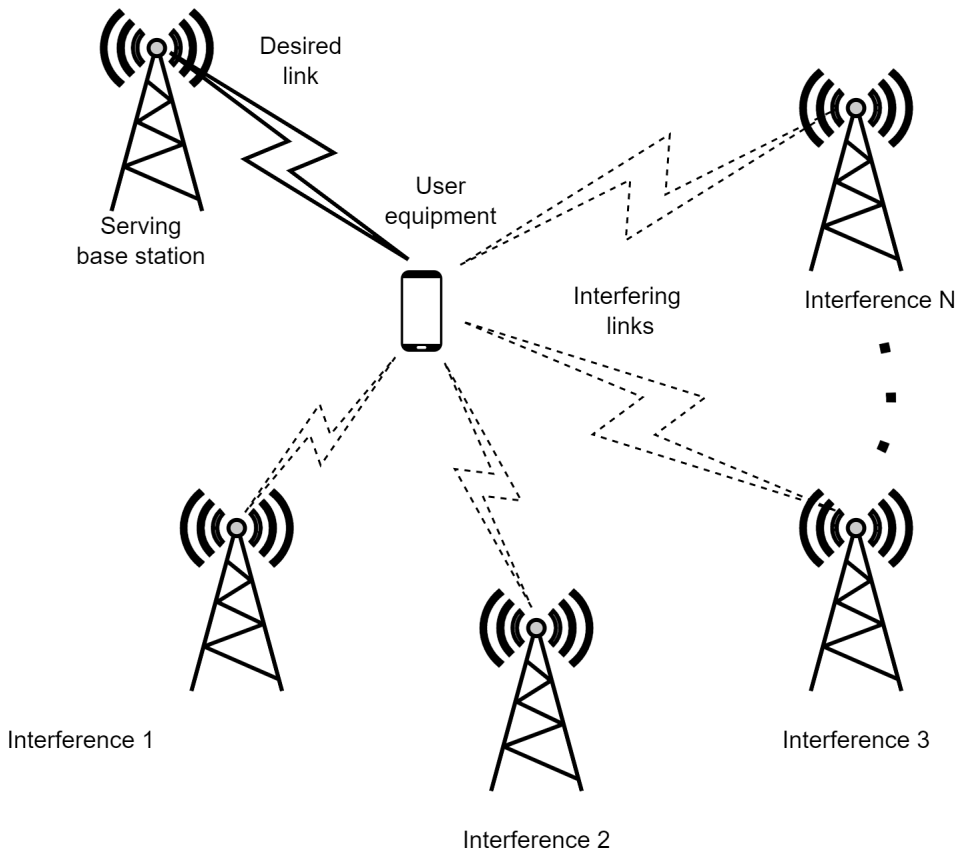

Figure 3.1. System model: desired link operates with N interferers.

URLLC transmissions typically happen in mini-slots that last less than 0.1 ms. This makes the transmission time interval (TTI) shorter and more flexible than the 1 ms TTI of LTE [22]. Since the coherence time of a conventional wireless transmission is much wider compared to the mini slot duration, the fading of the URLLC transmission may occur over multiple TTIs [23]. Within that coherence time, it is assumed that the transmitter

acquires sufficient knowledge about the desired channel state information (CSI). Also, it is expected that the user equipment (UE) will update the serving base station with the total interference values it has seen in the past. This will facilitate adequate knowledge about the interference at the transmitter, which will lead to a more efficient resource allocation in the downlink.

A brief packet of D bits with a target outage/block error rate (BLER) ε is transmitted by the desired transmitter. The power of the total interference signal is estimated by the transmitter using the EMD based prediction methods. Then the necessary resources (R) are subsequently allocated in accordance with the predicted interference power. To guarantee effective resource allocation, it is important to estimate the SINR ($\hat{\gamma}$) of the predicted signal as precisely as possible.

3.2 Prediction Model Training

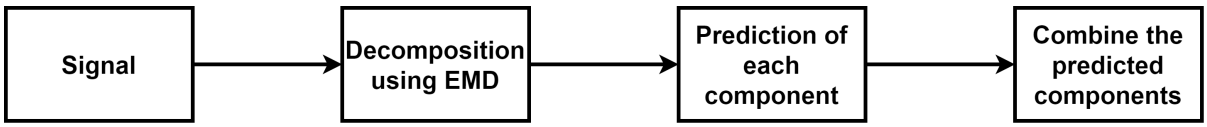

Figure 3.2. Proposed method for interference prediction.

The system model given above 3.1 has been addressed utilizing the theories discussed in the background Chapter 2, as shown in Figure 3.2. Initial interference signal generation has been done using presumptive INR parameters. The total interference signal that the mobile station experiences have then derived by adding all of the individual interference signals together as shown in

$$Totalinterference = \sum_{i=1}^{N} Interference(i), \tag{17}$$

where $Interference(i)$ is the interference from i^{th} interferer. The obtained total interference signal samples are split into two sets training samples and validation samples. The split can be performed using the predetermined training sample percentage value (ex 80%). The number of samples in the total interference signal is taken as T. The number of samples in training data and validating data are taken as N and M respectively. The total number of samples can be describe as

$$T = N + M. \tag{18}$$

Since developing a model to predict a linear (less random) signal over a nonlinear (random) signal is much easier and more accurate, IMFs and residual are used as inputs to the prediction models rather than directly feeding the signal as an input.

Then obtained training samples of total interference power N are decomposed into IMFs and residual using EMD. As described in the background section, the number of IMFs obtained after the EMD method is not a predetermined parameter and it only depends on the nature of the signal. For descriptive purposes, the total number of IMFs and residual is shown as L. Each IMF as well as the residual are fed into two forecasting algorithms: LSTM and ARIMA. Here, forecasting models have been trained individually

for each IMF and residual. The total number of models deployed in the LSTM method is equal to the sum of the numbers of IMFs and residuals (L). The same number of models will be used in ARIMA method as well (L). These two forecasting methods predict the next sample value based on the trained neural networks and time series forecasting methods as shown by

$$LSTM_{reconstructed} = \sum_{i=1}^{L-1} IMF_{LSTM(i)} + Residual_{LSTM}, \tag{19}$$

$$ARIMA_{reconstructed} = \sum_{i=1}^{L-1} IMF_{ARIMA(i)} + Residual_{ARIMA}, \tag{20}$$

where $IMF_{LSTM(i)}$ is the predicted value of IMF_i using the LSTM method, $IMF_{ARIMA(i)}$ is the predicted value of IMF_i using ARIMA method, $Residual_{LSTM}$ is the predicted value of residual using LSTM method, and $Residual_{ARIMA}$ is the predicted value of residual using ARIMA method.

Since there are M validating data points, the above one-step prediction method is repeated M times to obtain the predicted value set, as depicted in Figure 3.3. The advantage of iteratively predicting one future value at a time over predicting all M future values at once is that we can optimize the model by comparing the most recent predicted value to the actual value at each step. By always incorporating the most recent data points into model training, we can achieve higher prediction accuracy.

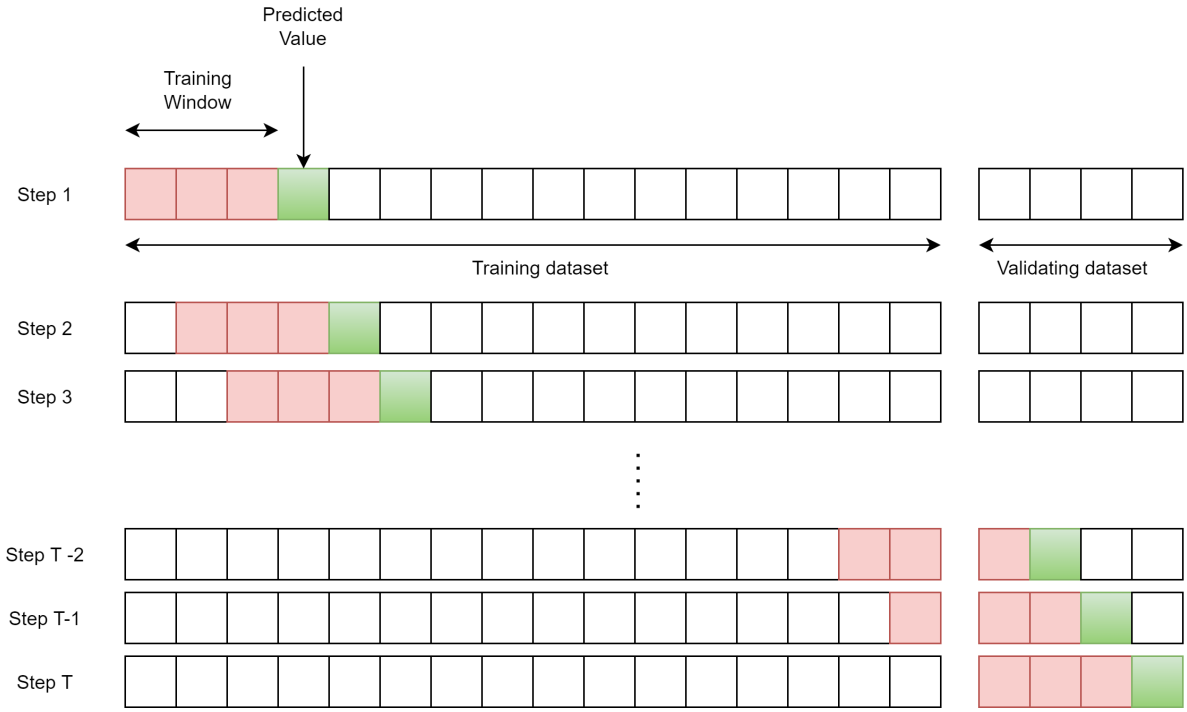

Figure 3.3. Training window selection over dataset.

According to the equations (19) and (20), the forecasted IMF and residual values are then added to obtain the final forecasted interference signal. At this time, the ARIMA reconstructed signal and the LSTM reconstructed signal will be available as anticipated signals.

$$LSTM_{reconstructed} = \sum_{t=1}^{M}(\sum_{i=1}^{L-1} IMF_{LSTM(i)}(t) + Residual_{LSTM}(t)), \tag{21}$$

$$ARIMA_{reconstructed} = \sum_{t=1}^{M}(\sum_{i=1}^{L-1} IMF_{ARIMA(i)}(t) + Residual_{ARIMA}(t)), \tag{22}$$

Apart from the decomposed signals, training samples of the total interference are also fed into the LSTM and ARIMA models individually and predictions are obtained, as shown below

$$LSTM_{Signal} = \sum_{i=1}^{M} Signal_{LSTM(i)}(t), \tag{23}$$

$$ARIMA_{Signal} = \sum_{i=1}^{M} Signal_{ARIMA(i)}(t). \tag{24}$$

In order to evaluate the performance of the EMD method, we compare the results of the reconstructed signal which we obtained from the EMD method with the straightforward prediction results obtained without using the EMD method. The above comparison can be done with the ARIMA model as well as with the LSTM model. The performance comparison is done by comparing the root mean squared error (RMSE) of the predictions. If the EMD results in a lower RMSE value compared to a simple prediction, it is considered as the desired output. Otherwise, parameter optimization and retraining of models are required. Then, an optimized model will be used to predict interference in a real-world scenario and radio resources will be allocated accordingly. The preceding process is depicted as a flowchart in Figure 3.4.

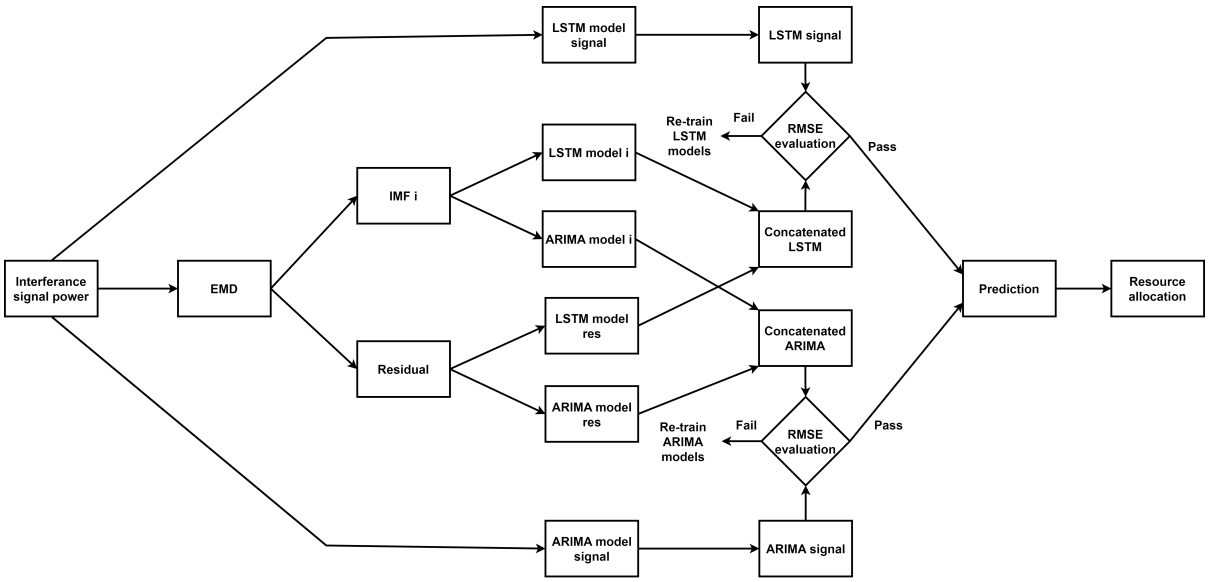

Figure 3.4. Model training procedure.

3.3 Resource Allocation

The predicted SINR $\hat{\gamma}$ will be calculated assuming that the power of the desired signal S is known using channel state information estimates, as given by

$$\hat{\gamma} = \frac{S}{\hat{I} + N_0}, \tag{25}$$

where $\hat{I}$ is the predicted interference power and N_0 is the normalized noise power.

According to finite block length theory [24], the amount of information bits D that can be dispatched with the given decoding error probability ε using R channel uses in additive white Gaussian noise (AWGN) channel can be calculated as (26)

$$D = RC(\hat{\gamma}) - Q^{-1}(\varepsilon)\sqrt{RV(\hat{\gamma})} + O(log_2 R), \tag{26}$$

$$C(\hat{\gamma}) = log_2(1 + \hat{\gamma}), \tag{27}$$

$$V(\hat{\gamma}) = \frac{1}{ln(2)^2}\left(1 - \frac{1}{(1+\hat{\gamma})^2}\right). \tag{28}$$

In the above (26), $C(\hat{\gamma})$ represents the Shannon capacity of AWGN under a finite block length regime and calculated as shown in (27). $V(\hat{\gamma})$ represents the channel dispersion and is measured in squared information units per channel, as shown in (28). Q^{-1} represents inverse of the Q-function. Using the preceding information, the channel utilization R can be estimated as below (29) [25]

$$R \approx \frac{D}{C(\hat{\gamma})} + \frac{Q^{-1}(\varepsilon)^2 V(\hat{\gamma})}{2C(\hat{\gamma})^2}\left[1 + \sqrt{1 + \frac{4DC(\hat{\gamma})}{Q^{-1}(\varepsilon)^2 V(\hat{\gamma})}}\right], \tag{29}$$

By rearranging the elements in (26), the block error rate can be calculated as (30)

$$\varepsilon \approx Q\left(\frac{RC(\hat{\gamma}) - D}{\sqrt{RV(\hat{\gamma})}}\right). \tag{30}$$

3.4 Performance Evaluation

Given that the performance of the LSTM and ARIMA methods is evaluated during prediction model training 3.2, this section examines the performance of resource allocation based on LSTM and ARIMA outputs.

First, we compute the SINR of predicted signal $\hat{\gamma}$ by (25) using predicted interference value $\hat{I}$ as an input. In 25 the power of the desired signal S is assumed to be known using the CSI. The normalised noise power is denoted by N_0 and is assumed to be one. Next, we compute the number of resources R required to transfer D bits with a target error probability ε. This resource allocation happens within one time step prior to the actual transmission $(t-1)$ as shown in timeline Figure 3.5.

After the transmission of D bits using allocated R number of resources at time t, we can obtain the actual total interference I that D number of bits had to gone through.

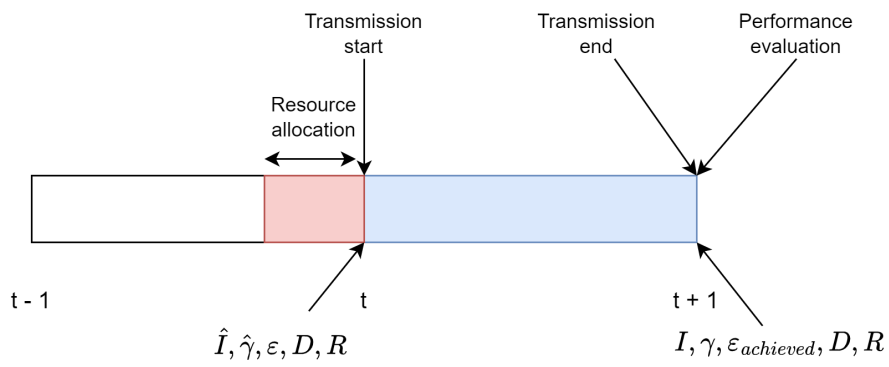

Figure 3.5. Resource allocation and performance evaluation timeline.

Therefore, we can obtain the actual SINR γ as shown below. It is important to note that, this time γ will be calculated using actual total interference I value.

$$\gamma = \frac{S}{I + N_0} \tag{31}$$

Since initially we have allocated the resources R based on target error probability ε, we can now obtain the achieved error probability using the same equation (30) but substituting predicted SINR value $\hat{\gamma}$ with the actual SINR value γ as shown below

$$\varepsilon_{achieved} \approx Q\left(\frac{RC(\gamma) - D}{\sqrt{RV(\gamma)}}\right). \tag{32}$$

Finally, we will be able to compare the target error probability with the achieved error probability. The above mentioned procedure can be iterated over different target error rate values to obtain different achieved error rate values.

The performance of the proposed scheme needs to be compaired against the baseline schemes. So the following two baseline schemes have been used [24].

- **Moving Average Based Estimation**

  This is a conventional weighted average based estimation procedure and it has been adapted to predict interference signal power as the first baseline scheme. Here, interference measured at time t is passed through the first order, low pass, infinite impulse response (IIR) filter and interference estimates are obtained according to the below equation

  $$\hat{I}_{t+1} = \alpha I_{t-1} + (1 - \alpha)\hat{I}_t, \tag{33}$$

  where, α is the forgetting factor (FF) of the filter ($0 < \alpha < 1$) and it determines weight given to the most recent measurement compared to previous one. Based on a simulation-based heuristic analysis, it has been determined that an $\alpha = 0.01$ is optimal for latency performance [26].

- **Genie Aided Estimation**

The genie aided estimator is considered as an optimal estimator. In this interference prediction scenario, it is assumed that the transmitter knows precisely what interference condition the transmitted signal will encounter. In other words, $\hat{I} = I$.

For the purpose of comparing the performance of the proposed model to that of the aforementioned baseline schemes, interference prediction is performed using the two baseline schemes. Therefore, the predicted interference values are $\hat{I}_{IIR}$ and $\hat{I}_{Genie}$. Then, using the procedure described above, the achieved outages for the same set of target outages will be obtained. Finally, conclusions were drawn by comparing the results of the proposed scheme to the results of the baseline scheme.

4 SIMULATION RESULTS AND DISCUSSION

First, the desired signal and total interference signal are generated using MATLAB. Next, the total interference signal is decomposed into IMFs and residual using the "emd" function in MATLAB. Then, the decomposed components are fed to the python platform. Future values of decomposed components are predicted using ARIMA and LSTM methods. The performance of the prediction models was also evaluated during the process. Next, the predicted interference values were forwarded for resource allocation. Finally, the overall performance of the proposed model is evaluated using the baseline criteria.

4.1 Signal generation

The following parameters have been used for the generation of desired signal and interference signal.

Table 4.1. Parameters used to generate desired signal and total interference signal.

Description	Value
Number of interferers N	5
SNR values of interferers	[5, 3, 0, -2, -5] dB
SINR value of desired signal $\bar{\gamma}$	20 dB
Number of samples considered	100
Channel model	Rayleigh block fading

MATLAB was used to generate the total interference and desired signal with the specified parameters as shown in Figure 4.1 and Figure 4.2.

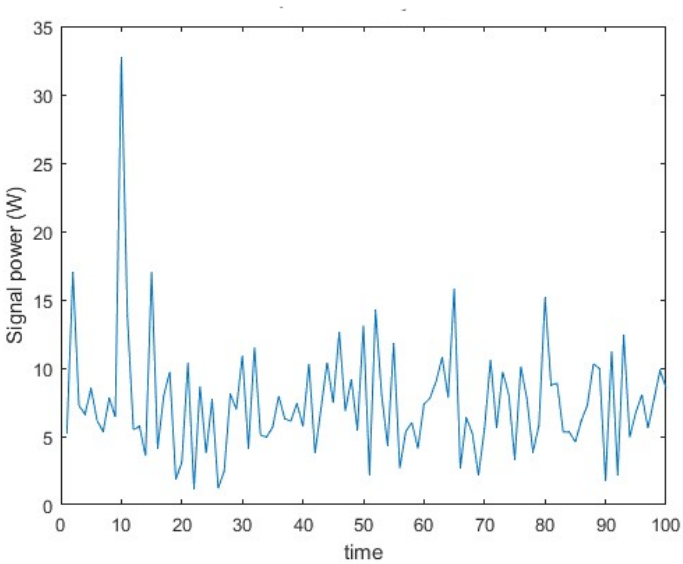

Figure 4.1. Total interference signal.

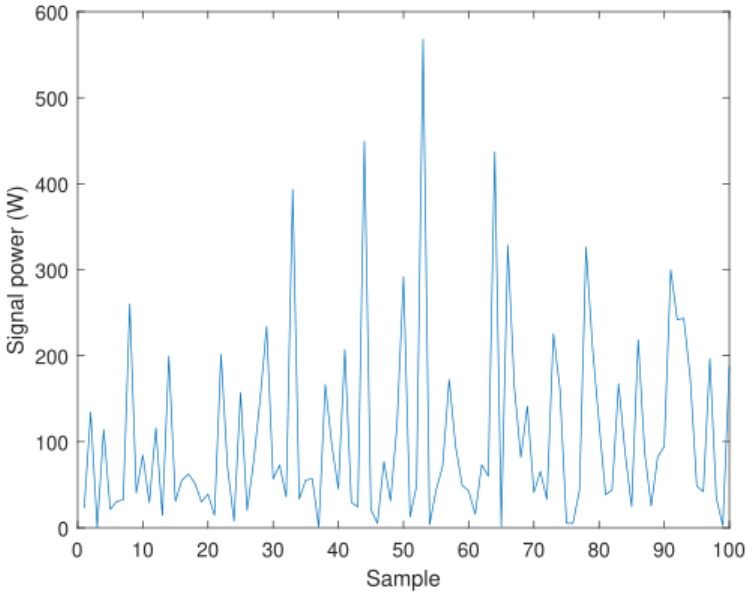

Figure 4.2. Desired signal.

4.2 EMD and Prediction Models

The total interference signal is decomposed using the MATLAB 'emd' function. The Python platform with Keras and Tensorflow backend is then used to develop LSTM and ARIMA models using the resulting IMFs and residuals. Since the desired signal is only necessary for taking the power, it is fed into python without EMD. For each IMF and residual, two models are trained. One model employs LSTM, and the other, ARIMA. The following two model training methods are used:

- Use same hyperparameters for all models.
- Use individually optimized hyperparameters for each model.

Finally, predicted components are summed up together to form the total predicted value, as shown in equation (19) and equation (20).

4.2.1 Same hyperparameters for all models

This model training method is utilized specifically to validate the EMD-based prediction accuracy in comparison to conventional prediction methods that do not use the EMD. Each model of IMFs, residual, and total interference signal are trained with the same hyperparameters.

Although the prediction models are trained according to individual training datasets of IMFs, residual, and total signal , the hyperparameters used for all these models remain the same. For example, all the models are trained using the following hyperparameters given in Table 4.2.

Table 4.2. Hyperparameters used to train all models.

Method	Hyperparameter	Value
LSTM/ARIMA	Training window	30
LSTM	Number of epochs	100
LSTM	Activation function	ReLU
LSTM	Optimizer function	adam
LSTM	Loss function	MSE
LSTM	Number of neurons in LSTM layer 1	100
LSTM	Number of neurons in LSTM layer 2	100
LSTM	Number of neurons in dense layer	1
ARIMA	Number of previous samples for AR p	30
ARIMA	Number of previous samples for MA q	0
ARIMA	Order of integration d	1

After training the models using the above parameters, predicted values of IMFs and residual are added together to form the total prediction, known as the reconstructed signal. Since we use two prediction methods (LSTM and ARIMA), we will get two reconstructed signals, known as $LSTM_{reconstructed}$ and $ARIMA_{reconstructed}$. Apart from the IMFs and residual components, the total interference signal is also predicted using LSTM and ARIMA. The names of the total interference signal prediction outputs are $LSTM_{signal}$ and $ARIMA_{signal}$. Using RMSE as the evaluation criterion, the performance of EMD-based prediction methods is compared to conventional prediction methods (without EMD). In each method, one RMSE value is obtained by considering reconstructed signal with the validation dataset of the total interference signal, and the other RMSE value is obtained by considering predicted value of the interference signal with the validation dataset of the total interference signal, as shown in below Figure 4.3.

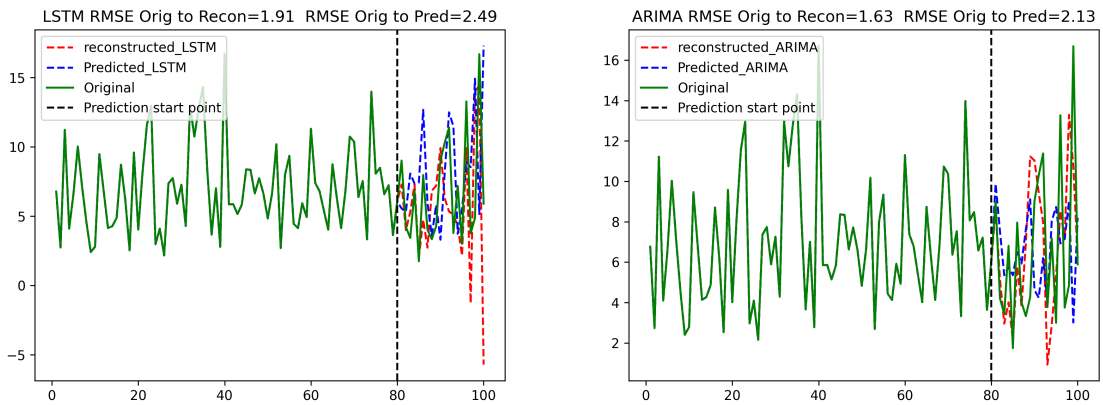

Figure 4.3. Performance evaluation of EMD with LSTM and ARIMA

As shown in the figures, the RMSE value of the reconstructed signal in both the LSTM and ARIMA methods is less than the RMSE value of the predicted signal. Therefore,

it suggests that EMD-based prediction methods outperform conventional prediction methods.

4.2.2 Individually optimized hyperparameters for each model

In this method, each model of IMFs and residual are trained using a unique set of hyperparameters that are optimized for a given dataset. When we consider the LSTM method, the hyperparameters used to train one IMF are different from the other IMF. It results in having different LSTM prediction models for each IMF and residual. Therefore, training of each LSTM model needs to be done individually using the required hyperparameters. The ARIMA method also follows the same procedure described above for the LSTM method. The reason for deploying different models for each IMF is that the linearity of the IMF increases as the IMF number increases. So, if we use the same hyperparameters for every model, more liner models will have over-fitting issues and more random signals will have under-fitting issues. Also, the model training time can be drastically reduced by using relaxed hyperparameters for higher IMFs. As an example, using a small $trainingwindow$ for higher IMFs will result in less training time and we can still obtain highly accurate predictions.

In LSTM, we used a sequential model with two LSTM layers and one dense layer. The developed LSTM models are compiled with the optimizer "adam" and a loss function of "mean squared error". The loss function measures how well the model performs during training. The optimizer is used to improve the performance of the model based on the output of the loss function. Then the model training is carried out by introducing the training dataset and setting the number of epochs. An epoch is the number of times an entire dataset is iterated through a neural network forward and backwards. The optimized hyperparameter set that we used to train LSTM models is included in Table 4.3.

Table 4.3. hyperparameters used for LSTM models.

Predicting element	**window**	**Epochs**	**LSTM 1 N**	**LSTM 2 N**
IMF 1	25	90	70	70
IMF 2	30	80	60	60
IMF 3	35	70	50	50
IMF 4	40	60	40	40
IMF 5	45	50	30	30
IMF 6	50	40	20	20
Residual	30	20	10	10

As shown in Table 4.3, as the IMF number increases, we can relax some training parameters such as: $Epochs$, $LSTM\,1\,N$, and $LSTM\,2\,N$. This is possible due to the randomness of the IMF signal reducing as the IMF number increases resulting prediction task being easier. However, the $window$ size needs to be increased as the IMF number increases. By the nature of the EMD as the IMF number increases it will form signals

with long-term dependencies rather than short-term dependencies. To capture these long-term dependencies accurately and make accurate predictions, we would require a wider training window.

In ARIMA time series forecasting, we just need to input three parameters to the model. Those are

- Number of past samples used for current prediction p,
- Number of past errors used for current prediction q,
- Order of the integration d.

Table 4.4. Hyperparameters used for ARIMA models.

Predicting element	**p**	**d**	**q**
IMF 1	30	1	0
IMF 2	35	1	0
IMF 3	45	1	0
IMF 4	50	1	0
IMF 5	55	1	0
IMF 6	65	1	0
Residual	70	1	0

As shown in Table 4.4, we can increase the p value as the IMF number increases allowing it to capture long-range dependencies of higher IMFs. It can be justified by considering the ACF of the decomposed signals. As shown in the sub figures in Figure 4.4, it can be seen that the correlation of the samples increases as the IMF number increases.

We can get the intuition about the training model selection method for each IMF and residual by comparing the accuracy of the prediction between LSTM and ARIMA, as shown in Figure 4.5 and Figure 4.6.

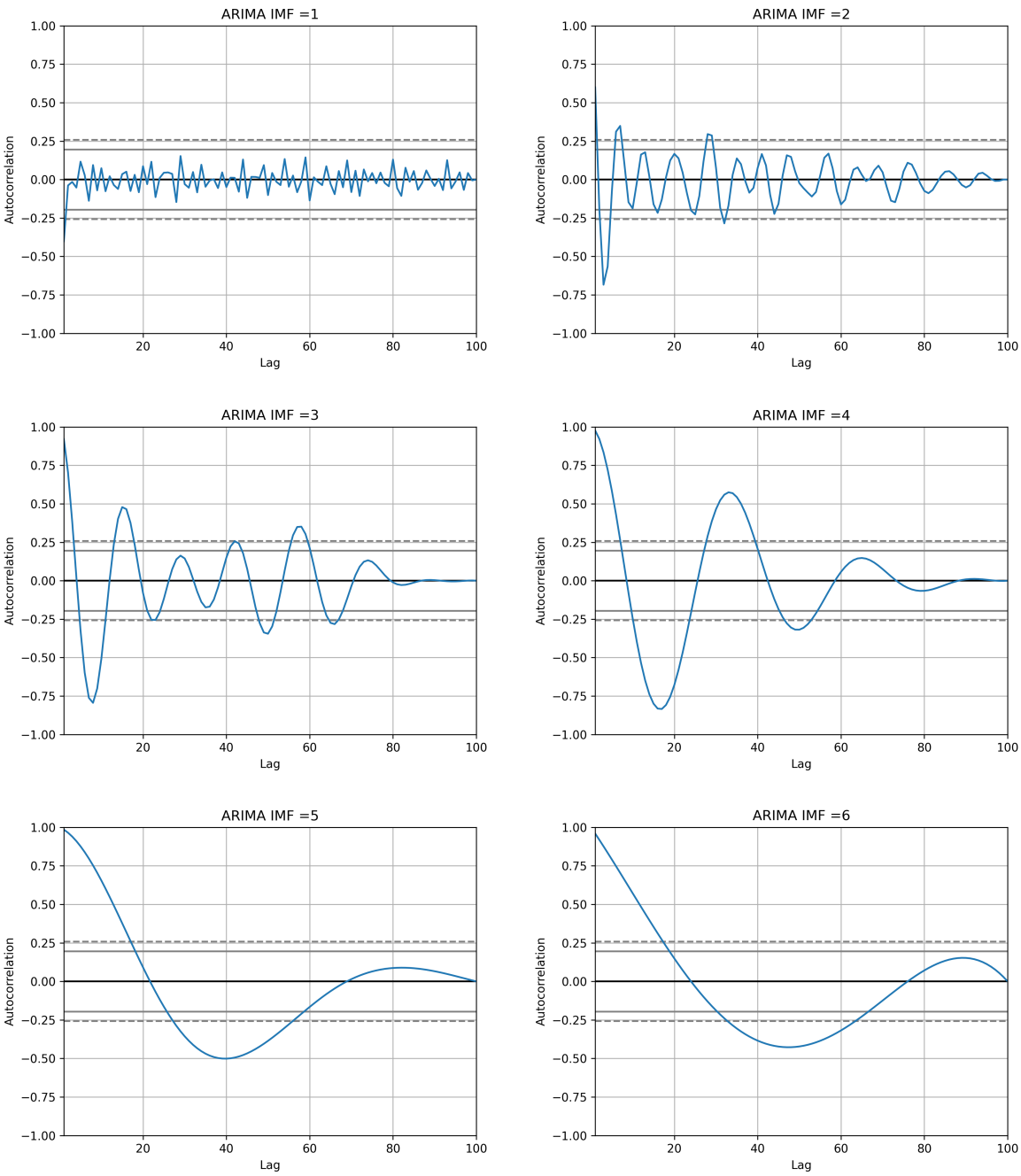

Figure 4.4. Auto correlation functions of IMFs.

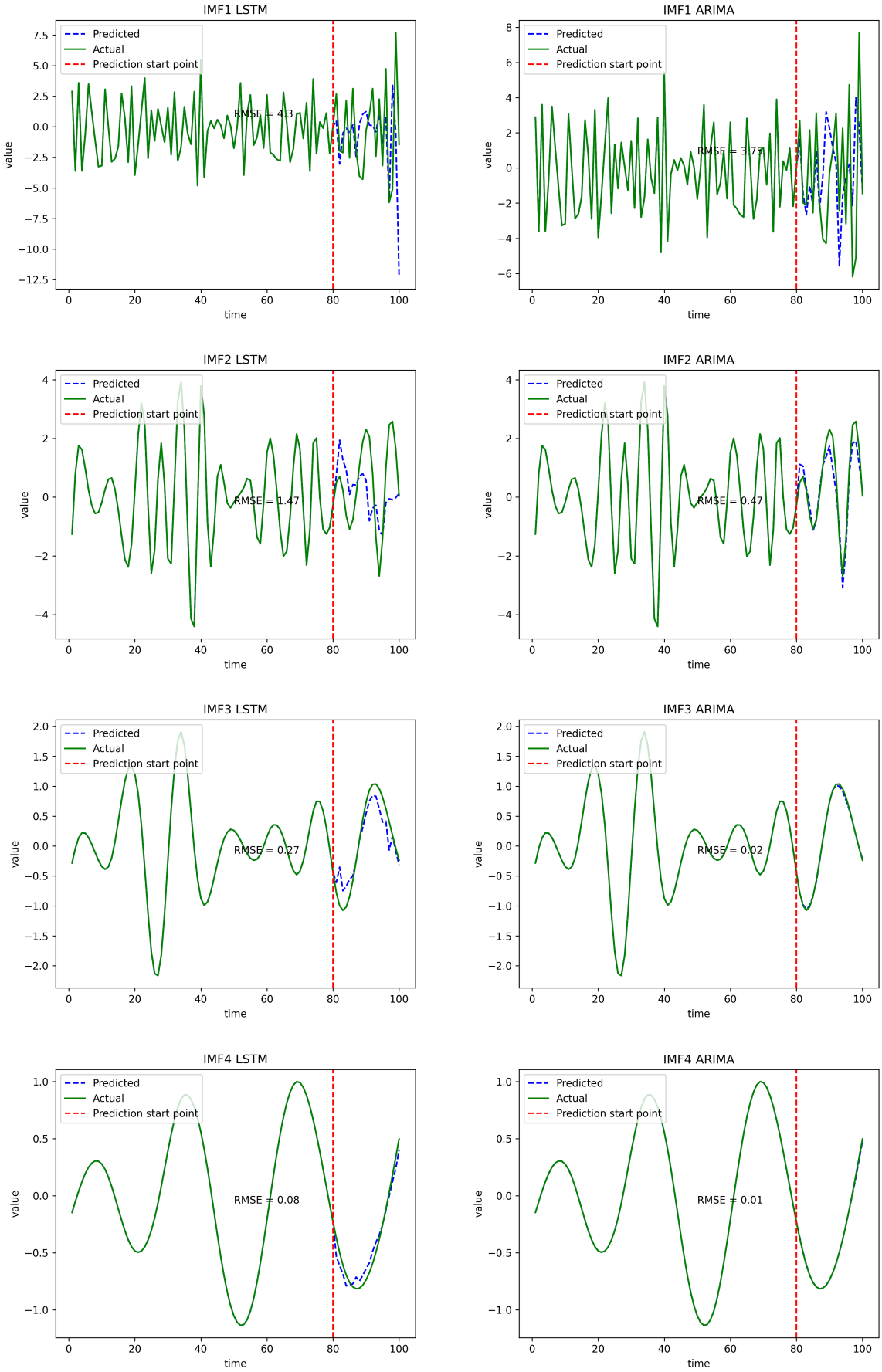

Figure 4.5. Prediction of each decomposed component using LSTM (left) and ARIMA (right).

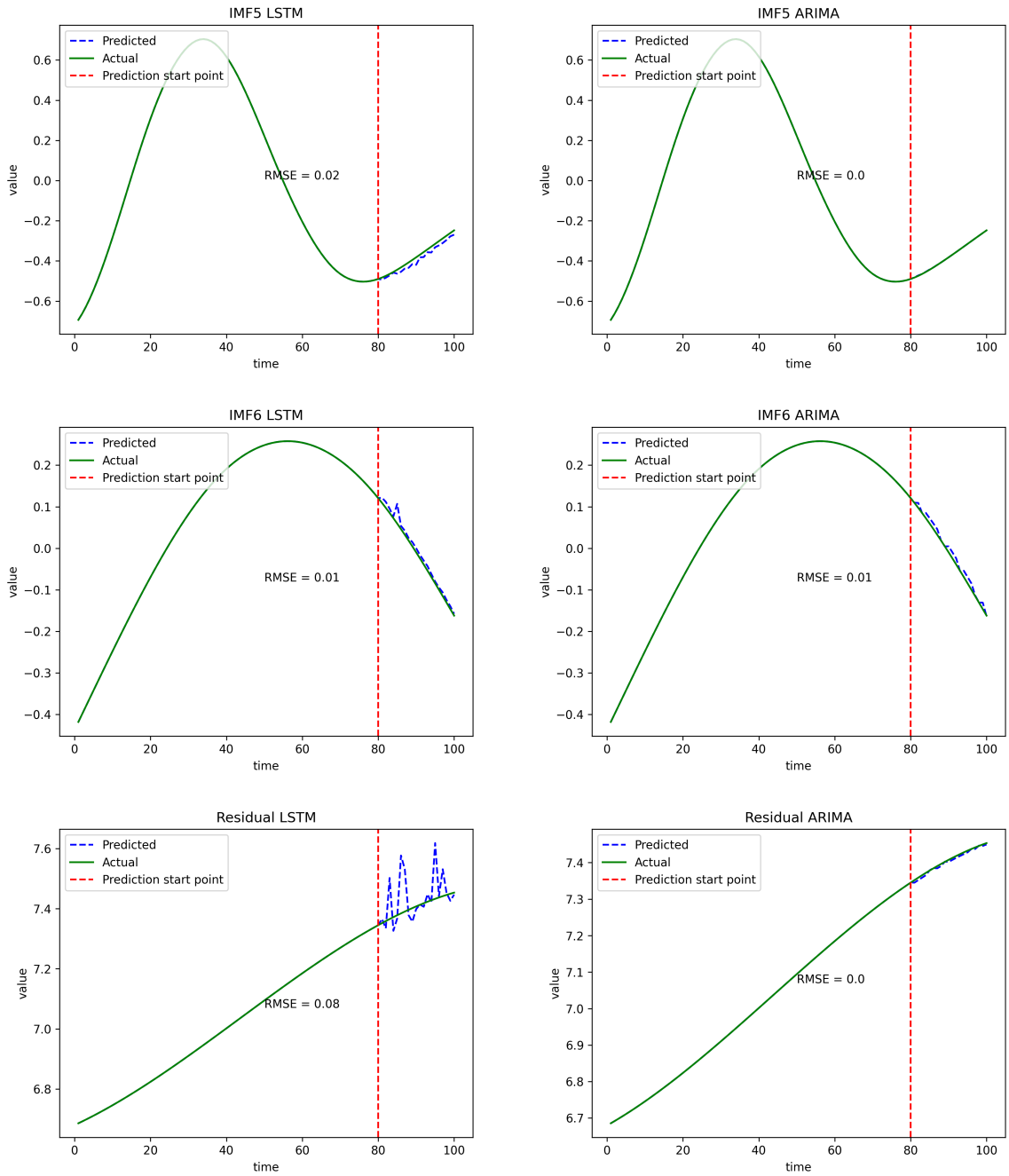

Figure 4.6. Prediction of each decomposed component using LSTM (left) and ARIMA (right).

4.3 Resource Allocation and Performance Evaluation

Based on predicted interference $\hat{I}$, resources are allocated for downlink channel. For the simulations, the interference signal and desired signal are generated as described in the section 4.1. The number of bits that we need to communicate in the downlink D is taken as 50. Achieved block error rate targets $\varepsilon_{achieved}$ were obtained by simulating different target error rate values $\varepsilon = (10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1})$.

The performance of the EMD based resource allocation methods are evaluated with the two baseline schemes described in 3.4.

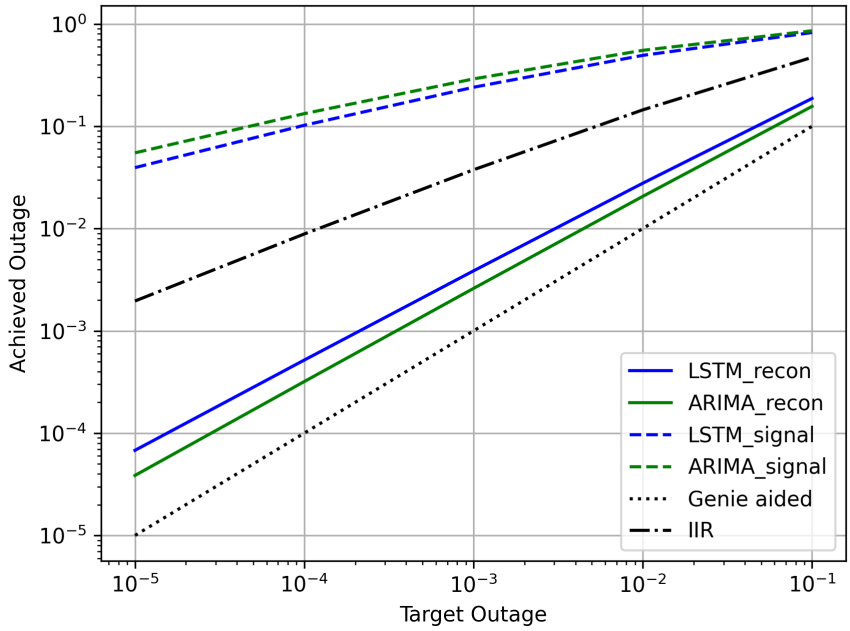

Figure 4.7. Achieved outage Vs target outage for different prediction methods.

Since the genie-aided estimator assumed that it already knew the achieved SINR, it could reserve the exact number of resources required to achieve the target outage, as shown in Figure 4.7. The curve generated by the genie-aided estimator is regarded as the optimal allocation of resources. Any curve that deviates less from the genie-aided curve can be considered an efficient allocation of resources, whereas efficiency decreases as the curve moves further from the genie-aided curve.

The performance of the IIR filter-based estimation is considered very poor, and it will only be able to achieve the block error rate (BLER) target of around 10%, as shown in Figure 4.7. The IIR based achieved outage curve shows considerable deviation from the expected genie-aided achieved outage curve. The achieved BLER targets of IIR filter-based estimation can be quite resource efficient only for eMBB services [1]. For URLLC services, stricter outage targets are anticipated.

We can observe that resource allocation without EMD utilizing only LSTM and ARIMA methods show poor performance compared to genie-aided resource allocation, but perform better than IIR-based resource allocation. In contrast, achieved outages using EMD-based prediction methods demonstrate better performance compared to IIR filter-based methods and methods that do not use EMD. The resource allocation method, which uses EMD and ARIMA for interference prediction, has allocated resources in a near-optimal way to meet target error rates, as shown in the $ARIMA_{recon}$ curve. Then, the second best resource allocation has been achieved with the EMD based LSTM method, as shown in the $LSTM_{recon}$ curve. It also proves the EMD based prediction methods were able to allocate resources more efficiently than the non-EMD prediction methods. The resource usage corresponding to the above outage targets is shown in

Figure 4.8. The genie-aided estimator has allocated an optimal minimal number of resources because it is assumed to have exact prior knowledge about the interference. Our proposed EMD-based prediction schemes have been allocated to a near-optimum performance, which assigns a lower number of resources compared to state-of-the-art interference prediction schemes.

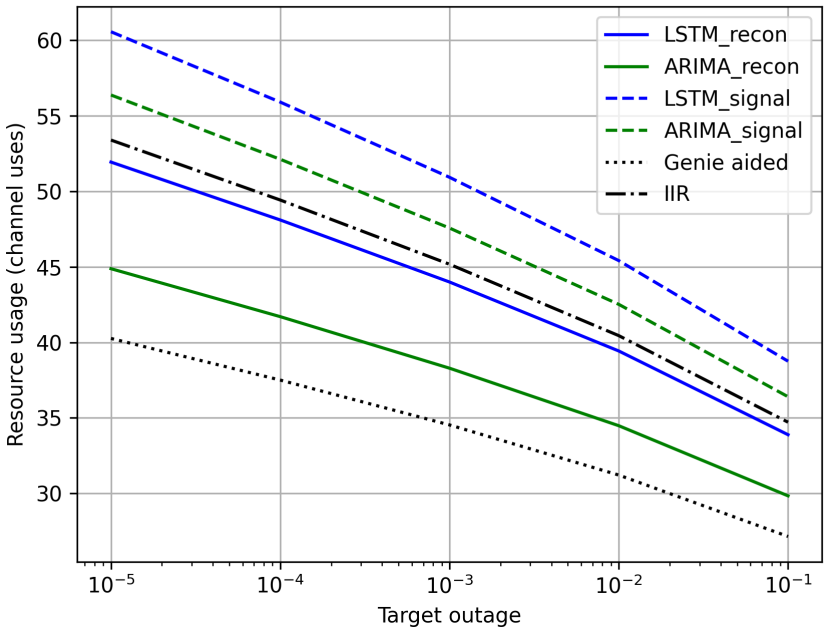

Figure 4.8. Resource usage Vs target outage for different prediction methods.

5 CONCLUSION AND FUTURE WORK

In this thesis, predictive resource allocation for URLLC services was investigated. The desired downlink at present of N interfering links was considered. We used the EMD algorithm to decompose the total interference signal power into IMFs and residual. This decomposition helped us to precisely predict future interference values. Along with the EMD, two prediction algorithms were used to train the models, LSTM and ARIMA. While heading towards the final objective of efficient resource allocation, the performance of an EMD based hybrid prediction scheme was also evaluated. Based on the predicted interference values, downlink resources were allocated to withstand at presence of actual interference. Finally, the performance of the suggested scheme was compared to that of the two baseline schemes and was able to achieve near optimal performance.

In the problem formulation, we assumed that past interference signal powers observed by UE were reporteded back to the serving base station. So the predictions were made by the serving base station using collected data from the UE.

In this research predicted interference power was obtained by either adding up all LSTM decomposed predictions or ARIMA decomposed predictions as shown in equation (19) and equation (20), further research could be conducted to use above two methods interchangeably for each decomposed signal element. A new classification criteria can be employed to select an appropriate prediction model based on the characteristics of the decomposed elements. So the total prediction will consist of some predicted components using ARIMA and LSTM.

6 REFERENCES

[1] Agiwal M., Roy A. & Saxena N. (2016) Next Generation 5G Wireless Networks: A Comprehensive Survey. IEEE Communications Surveys & Tutorials 18, pp. 1617–1655.

[2] Ji H., Park S., Yeo J., Kim Y., Lee J. & Shim B. (2018) Ultra-Reliable and Low-Latency Communications in 5G Downlink: Physical Layer Aspects. IEEE Wireless Communications 25, pp. 124–130.

[3] IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond. `https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf`.

[4] Ji H., Kim W. & Shim B. (2018) New radio technologies for ultra reliable and low latency communications. In: TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 1620–1623.

[5] Soret B., Mogensen P., Pedersen K.I. & Aguayo-Torres M.C. (2014) Fundamental tradeoffs among reliability, latency and throughput in cellular networks. In: 2014 IEEE Globecom Workshops (GC Wkshps), pp. 1391–1396.

[6] Ji H., Kim Y., Lee J., Onggosanusi E., Nam Y., Zhang J., Lee B. & Shim B. (2017) Overview of Full-Dimension MIMO in LTE-Advanced Pro. IEEE Communications Magazine 55, pp. 176–184.

[7] Han S., I C.l., Xu Z. & Rowell C. (2015) Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G. IEEE Communications Magazine 53, pp. 186–194.

[8] Bockelmann C., Pratas N., Nikopour H., Au K., Svensson T., Stefanovic C., Popovski P. & Dekorsy A. (2016) Massive machine-type communications in 5g: physical and MAC-layer solutions. IEEE Communications Magazine 54, pp. 59–65.

[9] Schulz P., Matthe M., Klessig H., Simsek M., Fettweis G., Ansari J., Ashraf S.A., Almeroth B., Voigt J., Riedel I., Puschmann A., Mitschele-Thiel A., Muller M., Elste T. & Windisch M. (2017) Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. IEEE Communications Magazine 55, pp. 70–78.

[10] Sivalingam T., Ali S., Mahmood N.H., Rajatheva N. & Latva-Aho M. (2021) Deep learning-based active user detection for grant-free SCMA systems. In: 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 635–641.

[11] Sivalingam T., Ali S., Huda Mahmood N., Rajatheva N. & Latva-Aho M. (2021) Deep neural network-based blind multiple user detection for grant-free multi-user shared access. In: 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–7.

[12] Padilla C., Hashemi R., Mahmood N.H. & Latva-Aho M. (2021) A Nonlinear Autoregressive Neural Network for Interference Prediction and Resource Allocation in URLLC Scenarios. In: 2021 International Conference on Information and Communication Technology Convergence (ICTC), pp. 184–189.

[13] She C., Sun C., Gu Z., Li Y., Yang C., Poor H.V. & Vucetic B. (2021) A Tutorial on Ultrareliable and Low-Latency Communications in 6G: Integrating Domain Knowledge Into Deep Learning. Proceedings of the IEEE 109, pp. 204–246.

[14] Xu S., Chang T.H., Lin S.C., Shen C. & Zhu G. (2016) Energy-Efficient Packet Scheduling With Finite Blocklength Codes: Convexity Analysis and Efficient Algorithms. IEEE Transactions on Wireless Communications 15, pp. 5527–5540.

[15] Rhee I. (1998) Retransmission-based error control for interactive video applications over the Internet. In: Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No.98TB100241), pp. 118–127.

[16] Tian Z. (2021) Approach for Short-Term Traffic Flow Prediction Based on Empirical Mode Decomposition and Combination Model Fusion. IEEE Transactions on Intelligent Transportation Systems 22, pp. 5566–5576.

[17] Huang N., Shen Z., Long S., Wu M., Shih H., Zheng Q., Yen N.C., Tung C.C. & Liu H. (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454, pp. 903–995.

[18] Mahmud A. & Mohammed A. (2021) A Survey on Deep Learning for Time-Series Forecasting, Springer, Cham, chap. 01. pp. 365–392.

[19] Zhou B., He D. & Sun Z. (2006) Traffic modeling and prediction using ARIMA/GARCH model, Springer, Boston, MA, chap. 01. pp. 101–121.

[20] Brockwell P.J. & Davis R.A. (2016) Introduction to Time Series and Forecasting. Springer Texts in Statistics, Springer International Publishing. URL: `https://doi.org/10.1007/978-3-319-29854-2`.

[21] Le X.H., Ho H., Lee G. & Jung S. (2019) Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. Water 11, p. 1387.

[22] Sachs J., Wikstrom G., Dudda T., Baldemair R. & Kittichokechai K. (2018) 5G Radio Network Design for Ultra-Reliable Low-Latency Communication. IEEE Network 32, pp. 24–31.

[23] Mahmood N.H., López O.A., Alves H. & Latva-Aho M. (2021) A Predictive Interference Management Algorithm for URLLC in Beyond 5G Networks. IEEE Communications Letters 25, pp. 995–999.

[24] Polyanskiy Y., Poor H.V. & Verdu S. (2010) Channel Coding Rate in the Finite Blocklength Regime. IEEE Transactions on Information Theory 56, pp. 2307–2359.

[25] Anand A. & de Veciana G. (2018) Resource Allocation and HARQ Optimization for URLLC Traffic in 5G Wireless Networks. IEEE Journal on Selected Areas in Communications 36, pp. 2411–2421.

[26] Pocovi G., Soret B., Pedersen K.I. & Mogensen P. (2017) MAC layer enhancements for ultra-reliable low-latency communications in cellular networks. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1005–1010.