

Received January 29, 2020, accepted February 20, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977760

# Real-Time Quantile-Based Estimation of Resource Utilization on an FPGA Platform Using HLS

CHANAKA GANEWATTHA<sup>1</sup>, ZAHEER KHAN<sup>1</sup>, JANNE J. LEHTOMÄKI<sup>1</sup>,  
AND MARJA MATINMIKKO-BLUE<sup>1</sup>

Centre for Wireless Communications (CWC), University of Oulu, 90014 Oulu, Finland

Corresponding author: Chanaka Ganewattha (chanaka.ganewattha@oulu.fi)

This work was supported in part by the Business Finland through the MOSSAF Project, and in part by the Academy of Finland through the 6Genesis Flagship under Grant 318927.

**ABSTRACT** Hardware accelerated modules that can continuously measure/analyze resource (frequency channels, power, etc.) utilization in real-time can help in achieving efficient network control, and configuration in cloud managed wireless networks. As utilization of various network resources over time often exhibits broad and skewed distribution, estimating quantiles of metrics to characterize their distribution is more useful than typical approaches that tend to focus on measuring average values only. In this paper, we describe the development of a real-time quantile-based resource utilization estimator module for wireless networks. The intensive processing tasks run on the FPGA, while the command and control runs on an embedded ARM processor. The module is implemented by using high level synthesis (HLS) on a Xilinx's Zynq-7000 series all programmable system on chip board. We test the performance of the implemented quantile estimator module, and as an example, we focus on forecasting congestion with real frequency channel utilization data. We compare the results from the implemented module against the results from a theoretical quantile estimator. We show that with high accuracy and in real time, the implemented module can perform quantile estimation and can be utilized to perform forecasting of congestion in wireless frequency spectrum utilization.

**INDEX TERMS** 5G, channel resource allocation, wireless channel congestion, forecasting, FPGA, generalized extreme value theory, high level synthesis, URLLC, Xilinx, ZedBoard.

## I. INTRODUCTION

A wide range of ultra-reliable and low latency communication (URLLC) applications, such as autonomous driving, robotics and industrial automation [1], require modules that can perform real-time analytics of key metrics relating to resource utilization in the network. These modules can help facilitate efficient resource allocation decisions in cloud managed wireless networks. Advances in software/hardware technologies allow wireless operators to deploy such analytics modules which are dedicated for measurement/collection of data inside their various network elements like access points/base stations (APs/BSs) [2], [3]. For example, Cisco System's Meraki Cloud Controller (MCC) utilizes a third radio dedicated for continuously monitoring the surroundings in each of its Meraki APs to improve the system-wide

performance [3]. Moreover, there are several works, which have also shown the benefit of deploying such data analytics modules in a wireless network as a separate sensor node. For example, use of dedicated sensors that act as analytics modules for fifth generation (5G) wireless systems have been proposed in [4], [5]. The dedicated modules can be utilized to collect real time resource (frequency spectrum, power, and etc.) utilization related data analytics that is given as input to cloud-based network resource controllers for efficient network control, configuration and management. Moreover, using hardware accelerated modules in an AP or as a dedicated sensor at a network's edge solves the problem of sample transfer overhead. Sending only limited processed data that indicates important information relating to the estimated performance metric of interest makes more sense than sending large volumes of streaming data to the controller.

The application of real-time analytics modules to wireless systems has been so far limited in the sense that most of

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

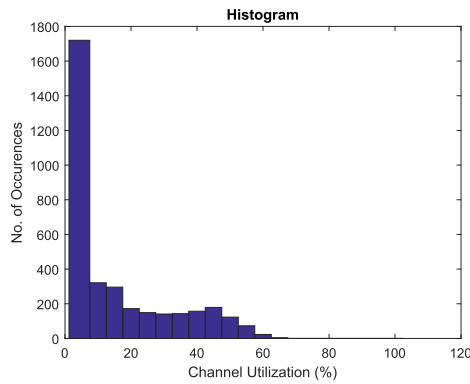


FIGURE 1. Distribution of wireless CU data.

the existing solutions have been developed based on average values of the metrics of interest. However, average values alone are not a good measure as distributions of various types of network data sets exhibit often skewed distributions and average alone does not capture the variations in the data correctly. To provide an example, in Fig. 1 we illustrate wireless frequency channel utilization (CU) data distribution using a histogram for a single day. The CU data was collected over an unlicensed channel by us in the University of Oulu. It can be seen from the figure that the distribution is right skewed. Approximating the CU data distribution with normal distribution using average often leads to inaccurate results. Estimating high quantiles of CU data distribution or distributions of other similar metrics of interest is more useful than only measuring mean values [6]. Incorporation of real time quantile estimation modules at the network edge can help in characterization and prediction of resource utilization load [7]. This can facilitate a network resource controller to take smart and proactive resource allocation decisions in real-time which in turn would enable it to deliver the guaranteed quality of service and also scaling the network to fulfill service level agreements with respect to various performance metrics.

In this paper, we use high level synthesis (HLS) to develop a real-time quantile estimator connected via an AXI4-Stream interface to the accelerator coherency port (ACP) of the ARM central processing unit (CPU) in the Zynq-7000 All Programmable System on Chip (AP SoC) device. We present the system design of the module, its implementation using HLS and its performance evaluation using real wireless CU data. The main contributions of our work can be summarized as follows.

- A real-time quantile estimation solution using FPGA which can process streaming samples and estimate quantiles which can then be used for making resource allocation decisions on a cloud controller server. The module contains three intellectual property (IP) cores which are created with HLS and integrated in a Zynq design using AXI4-Stream protocol.
- The data processing IP cores such as computation of histogram, cumulative sum, inverse cumulative sum using linear interpolation are implemented on the

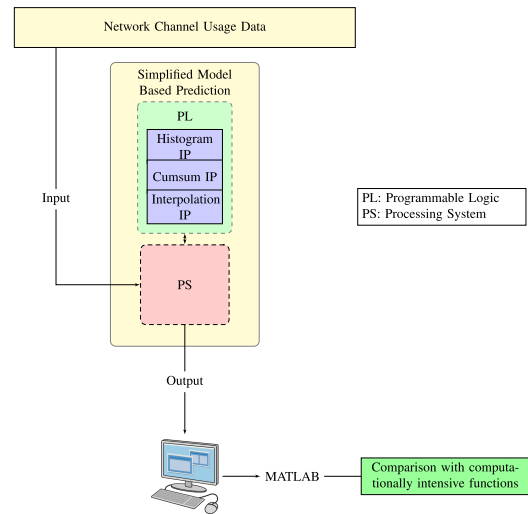


FIGURE 2. System model.

programmable logic (PL) part of a modern AP SoC device. The processing system (PS) of the AP SoC device is used for command and control. The system is based on interrupts with direct memory access (DMA) controller which transfers the data from the main memory to the IP cores and returns the processed data from the IP cores to the main memory with minimal CPU intervention. Fig. 2 gives an overview of the implemented solution.

- Our implemented solution can be used for estimating quantiles in real time of streaming data samples exhibiting different distributions, such as normal, Pareto, and generalized extreme value distribution. As an example, we test the performance of the implemented quantile estimator module with extreme value distribution for maxima wireless CU data collected over an unlicensed band in the University of Oulu. With the application of extreme value theory and quantile estimation, we show that our proposed system can accurately estimate quantiles. We also show that the estimated high quantiles can be used to forecast frequency spectrum resource utilization of a network under congestion. By forecasting the high quantiles of CU, which are directly related to the level of congestion of a network, we can ensure a certain probability of the service level for that network.
- We use a ZedBoard which is equipped with a Xilinx Zynq-7000 AP SoC for the practical implementation of the analytics device. We compare the quantiles estimated from our solution with the quantiles computed by Matlab using its built-in functions and we verify the accuracy of our solution.

It is worth noting that although CPUs and graphics processing units (GPUs) can be used for processing of wireless data, a system built with CPUs and GPUs is bulky, expensive and power hungry. Utilization of an FPGA makes more sense due to the lower cost and lower power consumption. FPGAs can be used to implement complex operations, parallelized functions and pipelined designs efficiently.

Very complex operations can be implemented to work within few clock cycles using the parallel resources which are readily available in the FPGA. This nature of FPGAs is very well suited for pipelining and real-time streaming kind of data processing applications. Also, reconfigurability of FPGAs to tailor its processing for a specific application makes them well suited to be applied across a wider range of applications. In our work, for the calculation of the quantiles of interest, instead of calculating all the quantiles, we can calculate only the quantiles of interest (FPGA can be reconfigured to find any arbitrary quantiles). Hence, we implement our solution in an FPGA.

The rest of the paper is organized as follows. In Section II, we present an overview to the related literature. In Section III, we provide a background to the CU and extreme value theory. In Section IV, we provide details in the implementation of the data analytic IP cores in the FPGA. Then, in Section V, we provide an evaluation of the results of the proposed system. Finally, Section VI gives the conclusion remarks of our work.

## II. RELATED WORK

Traditionally, much research emphases has been on non-real time data analytics where the wireless data is collected on computers or servers and processed in non-real time [8]. This approach leads to limitations in performance due to the inaccuracies in using past knowledge in optimizing real-time systems. Gaps in samples cause statistical inaccuracies leading to erroneous decision making. Due to the latency incurred in processing large data sets in the server, the system is unable to respond to the sudden needs in the network. Therefore, such wireless analytics systems are not useful for networks supporting real time application scenarios. Various types of non-real time data for network planning and management has been utilized in [9]–[11].

Recently, for 5G and beyond wireless networks, there has been more focus on the usage of real time data analytics techniques [12], [13]. It has been identified that utilizing real-time data analytics and data mining in wireless communication networks would enable dynamic network management, traffic engineering, radio access selection and network traffic steering [14]. It would also improve overall network performance enabling a 5G and beyond network to meet the stringent key performance indicators (KPIs) [14]. Towards this end, the latest 3GPP specification [15] has introduced a dedicated function called Network Data Analytics Function (NWDAF) which is responsible for providing network data analysis information upon request from other network functions. As an example, information on the traffic load level of a certain network slice could be provided upon a request by a certain other network function such as Policy Control Function (PCF) or Network Slice Selection Function (NSSF) [14]. The work in [16] presents a brief survey of various models proposed by the research community which focuses on the application of data mining and analytics in 5G and beyond networks.

Extreme value theory is a statistical data analysis tool which has been used by several research works to successfully forecast time series data [17]–[20]. The work presented in [17] introduces an application of extreme value theory in predicting cyber attacks using network data. It proposes a methodology for the prediction of attack rates in the presence of extreme values in observed data. According to the authors, the cyber attacks usually are assumed to exhibit extreme value phenomenon. Using an integration of time series theory for short term predictions and extreme value theory for long term predictions, the authors have developed a model which can predict cyber attack rates one hour ahead of time with practical prediction accuracy. In [18], a novel approach is presented to detect outliers in high throughput numerical time series without assuming any distribution for the input data and without manual thresholds setting. The authors introduce two methods, stationary peaks over threshold (SPOT) for streaming data with stationary distribution and drift SPOT (DSPOT) for streaming data with some drift. Peaks over threshold (POT) approach and maximum likelihood estimation have been used by the authors in [18] to estimate the required parameters for generalized Pareto distribution (GPD). The authors show few example applications of their anomaly detection algorithm in intrusion detection, magnetic field measurements and stock prices. In [21], a method of thread assignment in multi-core/multi-threaded processors using extreme value theory is presented. The authors use POT method to obtain the GPD for the thread assignment problem. The parameter estimation for the GPD is done using maximum likelihood estimation. The authors use a method called sample pruning, which reduces the time for statistical analysis and introduces a methodology called sample pruning POT (SP-POT) which is claimed to reduce the analysis time by eightfold. The authors declare that the introduced thread assignment performs close to optimal.

In applications where real-time data analytics are required, hardware acceleration using FPGAs has been the better choice [22]. FPGAs provide massive performance with its reconfigurable parallel resources than CPUs and GPUs and consume a fraction of power of a CPU or a GPU [23]. Due to this attractive property of FPGAs, they are often favored to be utilized for implementing computationally intensive statistical signal processing algorithms. The author of [24] presents a histogram based probability density function estimation using FPGAs. Cumulative distribution function is computed in real time for the input data and important information like centiles which are used in quality of service oriented decisions in communication systems, are calculated from the probability density function. The estimator is built using access pattern memory and priority encoders to reduce latency and increase the performance. The author claims that the proposed architecture uses minimal amount of hardware resources and area in the FPGA chip. In [25], FPGA based bandwidth selection for kernel density estimation is presented. An algorithm called plug-in is used for the estimation

of univariate kernel density estimation. The authors have used different architectural optimizations using the parallelism in FPGA to speed up the calculation. They have implemented a faster version of division operation using multiplications and divisions. For exponent and logarithmic calculations, coordinate rotation digital computer (CORDIC) algorithm has been used. The authors claim an average speed up by about 32 times compared to a CPU with much higher power efficiency than a CPU.

The problem of real-time quantile based estimation of resource utilization on an FPGA platform has not been investigated in the literature so far. The methodology proposed in this paper is based on the research findings of the first author's master's thesis [26]. We have implemented the proposed solution in a Xilinx's Zynq-7000 series FPGA using HLS. The intensive data processing tasks run on FPGA while the embedded ARM processor takes care of the command and control of the system. The proposed architecture is highly suitable for real-time streaming type of data. We use extreme value theory to describe the mathematical concept behind our solution and we show why quantile estimates make sense in a congested scenario of a network resource. To the best of our knowledge, this is the first time such a study on real-time quantile estimation of resource utilization using FPGAs has been done.

### III. THEORY

#### A. PHYSICAL LAYER CHANNEL: BACKGROUND

Wireless CU is a measure of how much of the available air time is utilized by wireless networks and is typically expressed as a percentage. CU is a key metric as it can be used to assess the health of a wireless network at physical layer. For example, in *Wi-Fi* networks, CU can be used to assess the impact on the user experience relating to various applications running on various mobile devices and laptop computers. If the CU measured is at a higher value ranging from 70% to 90%, the user experience can be severely affected. We use CU samples as test data to verify the validity of our implemented quantile estimation module. We compare the estimated quantiles with a theoretical model approach based on extreme value theory.

CU data is directly obtained by processing IQ samples using real-time FPGA module implemented by us in [27]. For the accurate detection of the signal [27], we estimate the noise floor and then set a detection threshold appropriately as a scaled value of the estimated noise floor. When the received signal power  $I^2 + Q^2$  exceeds the threshold, we declare that the signal is present or otherwise the signal to be absent [27]. CU value thus calculated is given by [27]:

$$\alpha_t = \frac{L_F}{L_F + L_O} \quad (1)$$

where  $L_F$  denotes the number of instances in the time interval  $t$  in which the signal is present and  $L_O$  denotes the number of instances where the signal is absent. The number of samples in time  $t$  depends on the duration of the interval  $t$

and sampling frequency which would range from hundred of thousands to few millions.

#### B. EXTREME VALUE THEORY

The sequence of CU values given by  $\alpha_t$  which is measured as a certain process on a regular timescale can be considered as random variables which have a common distribution function. Let the maximum value drawn from  $n$  observations of the process over  $n$  time units be defined by  $M_n$ , then  $M_n = \max\{\alpha_{t+1}, \alpha_{t+2}, \dots, \alpha_{t+n}\}$ . This is also called the block maximum.

The extremal types theorem in classical extreme value theory states that there exists sequences of constants  $a_n > 0$  and  $b_n$  such that [28]

$$Pr \left\{ \frac{M_n - b_n}{a_n} \leq z \right\} \rightarrow G(z) \quad \text{as } n \rightarrow \infty \quad (2)$$

where  $G$  is a non-degenerate distribution function. Then extreme value theory states that  $G$  should belong to one of the families, Gumbel, Fréchet or Weibull distributions which are also called type I, type II and type III families of distributions respectively [28]. These distributions model different forms of tail behavior for the distribution function  $F$  of  $\alpha_i$ . For example, in Gumbel distribution, the density of  $G$  decays exponentially while for Fréchet distribution, it decays polynomially which correspond to different rates of decay in the tail of  $F$  relative to each other. These three families are reformulated to a single family of models having the distribution functions of the form given by [28],

$$G(z) = \exp \left\{ - \left[ 1 + \xi \left( \frac{z - \mu}{\sigma} \right) \right]^{-1/\xi} \right\} \quad (3)$$

which is defined on the set  $\{z : 1 + \xi(z - \mu)/\sigma > 0\}$ . Parameters  $\mu$ ,  $\sigma$  and  $\xi$  satisfy  $-\infty < \mu < \infty$ ,  $\sigma > 0$  and  $-\infty < \xi < \infty$  respectively. This is called generalized extreme value (GEV) family of distributions.  $\mu$ ,  $\sigma$  and  $\xi$  are called location parameter, scale parameter and shape parameter respectively. Fréchet and Weibull families correspond to the cases  $\xi > 0$  and  $\xi < 0$  respectively and  $\xi = 0$  corresponds to Gumbel family.

GEV distribution which is formed by the unification of the original three distribution families greatly simplifies the statistical implementation. The tail behavior can easily be determined by the inference on  $\xi$  and so that there is no need to assume individual extreme value family for a given set of data. Estimation of the parameters  $\mu$ ,  $\sigma$  and  $\xi$  is done by maximizing a log likelihood function using standard numerical optimization algorithms [28].

The above analysis of CU data makes sense since examining the collected CU data from [27] tells us that they have occasional spikes. These spikes or bursts occur due to high usage by a single user (due to for example downloading a large file) or high aggregated usage by multiple users. Due to this nature of the CU, the distribution of CU data is no longer normal. Instead, the bursts can result in heavily tailed distributions with large deviations from the mean or

the median of the CU data. Therefore, statistics like mean and standard deviation may not accurately predict the probabilities of the CU. Due to this reason, a forecast method based on mean and standard deviation may not be a viable solution. As CU data behave with extreme values due to the bursty nature, the theory of extreme value can effectively be applied in the modeling of wireless CU data.

**C. QUANTILE ESTIMATES BASED ON EXTREME VALUE THEORY**

The estimates of the extreme quantiles for the maxima series are obtained by the inversion of the GEV equation (3) which gives [28],

$$z_p = \begin{cases} \mu - \frac{\sigma}{\xi} [1 - \{-\log(1 - p)\}^{-\xi}], & \text{for } \xi \neq 0, \\ \mu - \sigma \log\{-\log(1 - p)\}, & \text{for } \xi = 0, \end{cases} \quad (4)$$

where  $G(z_p) = 1-p$ . The parameters,  $\sigma$ ,  $\xi$  and  $\mu$  are found by maximizing the log likelihood function as stated earlier.

**D. DERIVATION OF A SIMPLER VERSION OF THE GENERALIZED EXTREME VALUE THEORY FOR FPGA IMPLEMENTATION**

Estimation of quantiles using theoretical approach involves calculating logarithms and exponents which is complex and highly inefficient in terms of performance when implemented in hardware. Therefore, it is typical to use good approximations to the theory so as to ease the implementation aspect.

Due to the complexity in implementation of parameter estimation and equation computation of GEV for wireless CU data in a real-time system, we utilize the histogram which can readily be computed. Histogram with appropriate bin width is considered to be an accurate representation of the numerical data distribution. It is an approximation to the probability density function (pdf),  $p_X(x)$  of a random variable  $X$ . Let  $\mathcal{H}$  denote the histogram,  $I_i$  denote the  $i^{th}$  bin interval of the histogram,  $\underline{I}_i$  denote the minimum value and  $\bar{I}_i$  denote the maximum value of the  $i^{th}$  bin interval,  $\pi_i$  denote the frequency of the data values which lie within the  $i^{th}$  bin interval  $I_i$  and  $M$  be the number of histogram bins. Then, the bin interval  $I_i$  can be written as,

$$I_i = [\underline{I}_i, \bar{I}_i) \quad (5)$$

and the histogram can be denoted as,

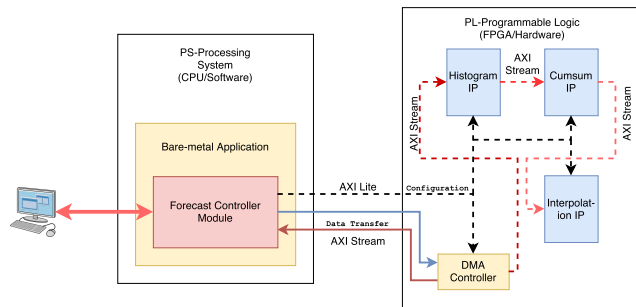
$$\mathcal{H} = \{(I_1, \pi_1), (I_2, \pi_2), \dots, (I_M, \pi_M)\} \quad (6)$$

From histogram, we obtain the distribution of cumulative sum which is used for the quantile estimation. Let  $\mathcal{C}$  denote the (empirical) cumulative distribution function of the input data samples. For  $i^{th}$  bin interval, we calculate,

$$\psi_i = \sum_{k=1}^i \pi_k \quad (7)$$

and obtain the distribution of cumulative sum as,

$$\mathcal{C} = \{(I_1, \psi_1), (I_2, \psi_2), \dots, (I_M, \psi_M)\} \quad (8)$$



**FIGURE 3. System block diagram.**

For the quantile estimation, we need to find the estimate of  $p = P(X > x)$ . As we are dealing with cumulative sum values, we compute  $P = p \times N$  where  $N$  denotes the total sum of histogram counts (in our case  $N = \psi_M$ ). We find the bin interval in the distribution of cumulative sum where  $P$  resides. Let this bin interval be  $I_j = [\underline{I}_j, \bar{I}_j)$ , then we can write,

$$P \in [\psi_j, \psi_{j+1}) \quad (9)$$

Then, using the linear interpolation, we find the quantile estimate value,  $Q \in [\underline{I}_j, \bar{I}_j)$  which is given by,

$$Q = \underline{I}_j + (P - \psi_j) \frac{(\bar{I}_j - \underline{I}_j)}{(\psi_{j+1} - \psi_j)} \quad (10)$$

Due to the reduced complexity of the proposed method, it is simple and easy to implement in hardware. Compared to the direct implementation of GEV theory, there is less arithmetic operations required and the method can readily be utilized for streaming type of data. Thus, the proposed system is very well suited for the hardware architecture in the FPGA. Real-time histogram generation and cumulative sum calculation for the streaming data can easily be implemented in FPGA. The quantile estimation method which is used for the wireless channel congestion forecasting can also be implemented with less latency. The downside of the proposed method can be the loss of accuracy for the implementation simplicity. However, we show that the degradation in the performance of the proposed method is minimal compared to the direct GEV implementation. In Section V, we thoroughly analyze the performance of both of the proposed method and the direct GEV implementation.

**IV. IMPLEMENTATION**

**A. SYSTEM OVERVIEW**

Fig. 3 depicts a high-level overview of the complete system architecture of our solution. The implementation is done using a Xilinx Zynq-7000 AP SoC in ZedBoard [29]. The Xilinx Zynq-7000 AP SoC contains 1) PS which features a dual core ARM Cortex A9 processor and 2) PL. All the other necessary peripherals like on-chip memory, external memory interfaces and I/O (Input/Output) peripherals and interfaces are included in the SoC [32]. Zynq-7000 AP SoC

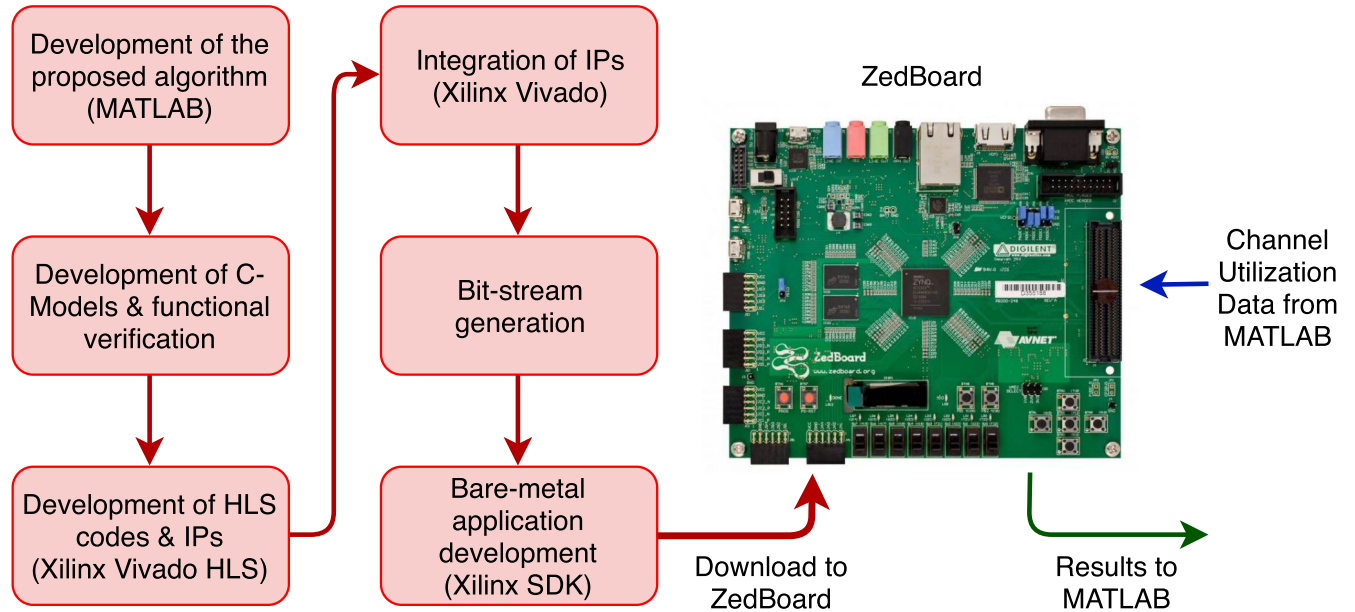


FIGURE 4. Steps involved in the implementation of the proposed system on a ZedBoard.

offers flexibility and scalability of an FPGA while delivering performance, power and ease of use typically associated with ASICs (application specific integrated circuit). Due to its flexibility in customization, PL can be configured according to the design requirement. PS of the AP SoC manages the PL part and the communication between them. The resulting outputs are obtained by a laptop connected to the ZedBoard.

Fig. 3 shows all the important functional blocks in the implemented system. The forecast system controller module resides in the PS and manages the data exchange between 1) the computer and PS 2) the PS and the PL. All the other functional blocks related to the forecast system reside in the PL. The system uses *Advanced eXtensible Interface (AXI)* bus specification [33] to exchange data among functional modules, and between the PS and the PL.

Due to the simplicity and flexibility of using HLS for IP core development [30], necessary functional modules have been designed/developed and corresponding IP cores have been generated using the tool, *Xilinx Vivado HLS* [30]. *Xilinx Vivado* has been used for the complete system integration and bit stream generation for hardware programming. The bare-metal (standalone) [31] application development has been done using the tool, *Xilinx Vivado SDK*. The key steps involved in the implementation of the proposed system are given in Fig. 4.

### B. FPGA ALGORITHM IMPLEMENTATION

The PL contains three functional modules (IPs), namely, *makehist* IP, *cumsum* IP and *incum* IP (interpolation IP). These IPs work in conjunction with the PS to evaluate the quantile estimate for a given CU probability. The IPs are arranged in a pipeline such that the processed data from one IP travel to the next IP in succession. The algorithms

implemented in each IP and their detailed functionalities are discussed in next sub sections.

#### 1) HISTOGRAM AND CUMULATIVE SUM IP MODULES

Algorithm 1 is used for the histogram calculation for the input CU data. As the CU data is streaming type, there is only single CU data value available at a certain time instance,  $i$ . Let  $D_i$  denote the CU data at time instance  $i$ ,  $\underline{I}_j$  and  $\bar{I}_j$  be the left and right bin edges at  $j^{\text{th}}$  bin interval  $I_j$ ,  $\pi_j$  be the histogram bin counter value at  $j$  and  $W$  be the CU data window for which the histogram is computed. The CU data value  $D_i$  is compared against the histogram bin interval,  $I_j = [\underline{I}_j, \bar{I}_j)$ . If  $D_i$  is inside the considered bin interval,  $\pi_j$  counter at location  $j$  is incremented by one and next CU data sample is taken for comparison. If  $D_i$  is not inside the  $j^{\text{th}}$  bin interval, it is compared with the next bin interval  $I_{j+1} = [\underline{I}_{j+1}, \bar{I}_{j+1})$ . This comparison is repeated until the CU data sample matches the corresponding bin interval in the histogram. The logic design which implements this functionality is given in Fig. 5. This process is continued for every CU data sample in the considered data window. Once all the CU data in the data window is processed, the computed histogram is given at the output. The IP implementing this algorithm is called the *makehist* IP. The timing, latency and resource utilization for this IP are given in Table 1, Table 2 and Table 3 respectively for a sample data window size of 64.

Algorithm 2 is used for the generation of cumulative sum. Let  $\psi_i$  represents the cumulative value at  $i^{\text{th}}$  location of the corresponding bin in the histogram. Cumulative value at  $i^{\text{th}}$  location is given by the sum of all the past histogram values up to the  $i^{\text{th}}$  bin. The algorithm calculates the sum over all the past histogram bin values for the  $i^{\text{th}}$  value and this process is repeated for all the histogram bins. Cumulative sum is given

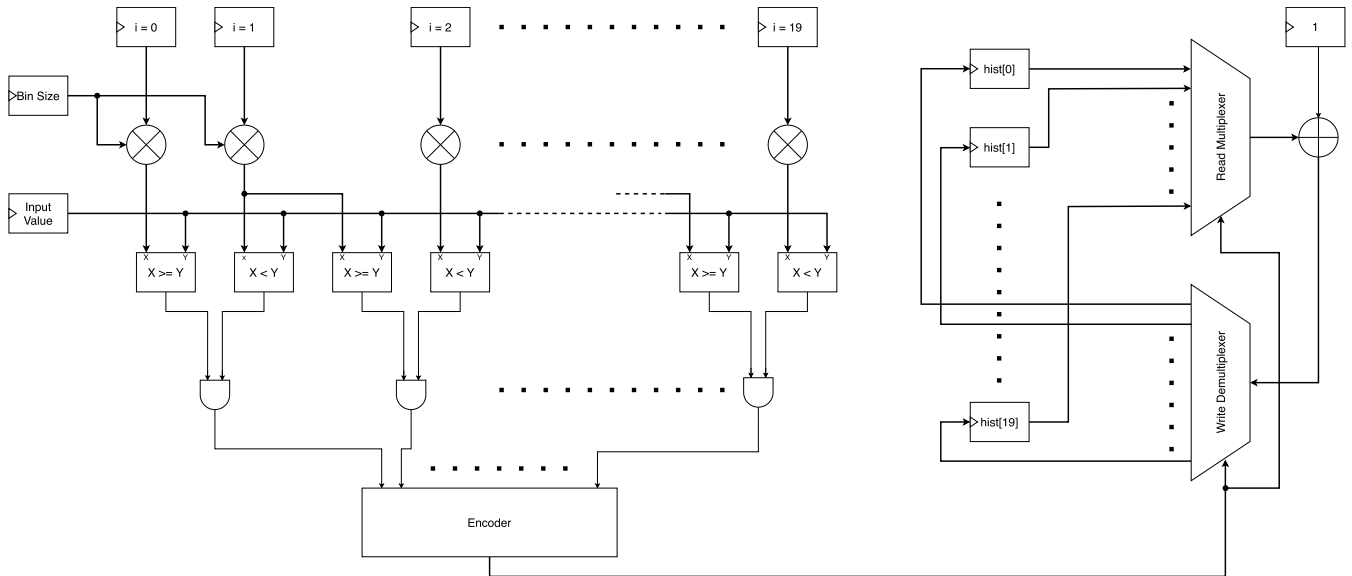


FIGURE 5. Logic design for the comparison and updating in histogram IP.

**Algorithm 1** Histogram Algorithm

```

1: Input: Data samples size of window  $W$ 
2: for each data sample  $D_i$  in  $W$  do
3:   for each  $j^{th}$  bin in histogram  $\mathcal{H}$  do
4:     if  $I_j \leq D_i < \bar{I}_j$  then
5:        $\pi_j \leftarrow \pi_j + 1$ ;
6:       break;
7:     end if
8:   end for
9: end for
10: Output: Histogram  $\mathcal{H}$ 
    
```

**Algorithm 2** Cumulative Sum Algorithm

```

1: Input: Histogram  $\mathcal{H}$ 
2:  $\mathcal{C} \leftarrow \mathcal{H}$ 
3:  $i \leftarrow 2$ ;
4: for each  $i^{th}$  bin in  $\mathcal{C}$  do
5:    $\psi_{i+1} \leftarrow \psi_{i+1} + \psi_i$ 
6: end for
7: Output: Cumulative sum  $\mathcal{C}$ 
    
```

TABLE 1. Timing summary of the generated IPs.

IP	Estimated Clock (ns)
Makehist	4.806
Cumsum	4.321
Invcum	8.510

as output at the end of this operation. The resulting logic design for this IP is given in Fig. 6. The IP implementing this algorithm is *cumsum* IP and its timing, latency and resource utilization information are given in Table 1, Table 2 and Table 3 respectively.

TABLE 2. Latency information of the generated IPs.

IP	Latency (clock cycles)	
	Min	Max
Makehist	36	288
Cumsum	35	35
Invcum	102	102

TABLE 3. Resource utilization of the generated IPs on Xilinx Zynq-7000 xc7z020clg484-1 device.

IP	Resource			
	BRAM_18K	DSP48E	FF	LUT
Makehist	0	0	1051	1409
Cumsum	0	0	319	484
Invcum	0	10	1663	2453

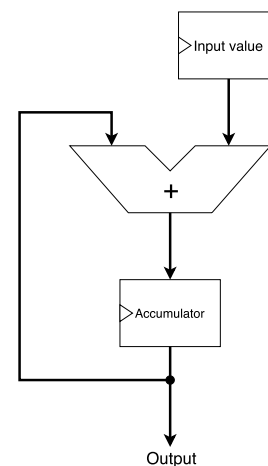


FIGURE 6. Logic design for the cumulative sum IP.

2) QUANTILE ESTIMATION IP

Algorithm 3 shows the steps in estimating the quantile for a given probability. As the cumulative sum is derived for

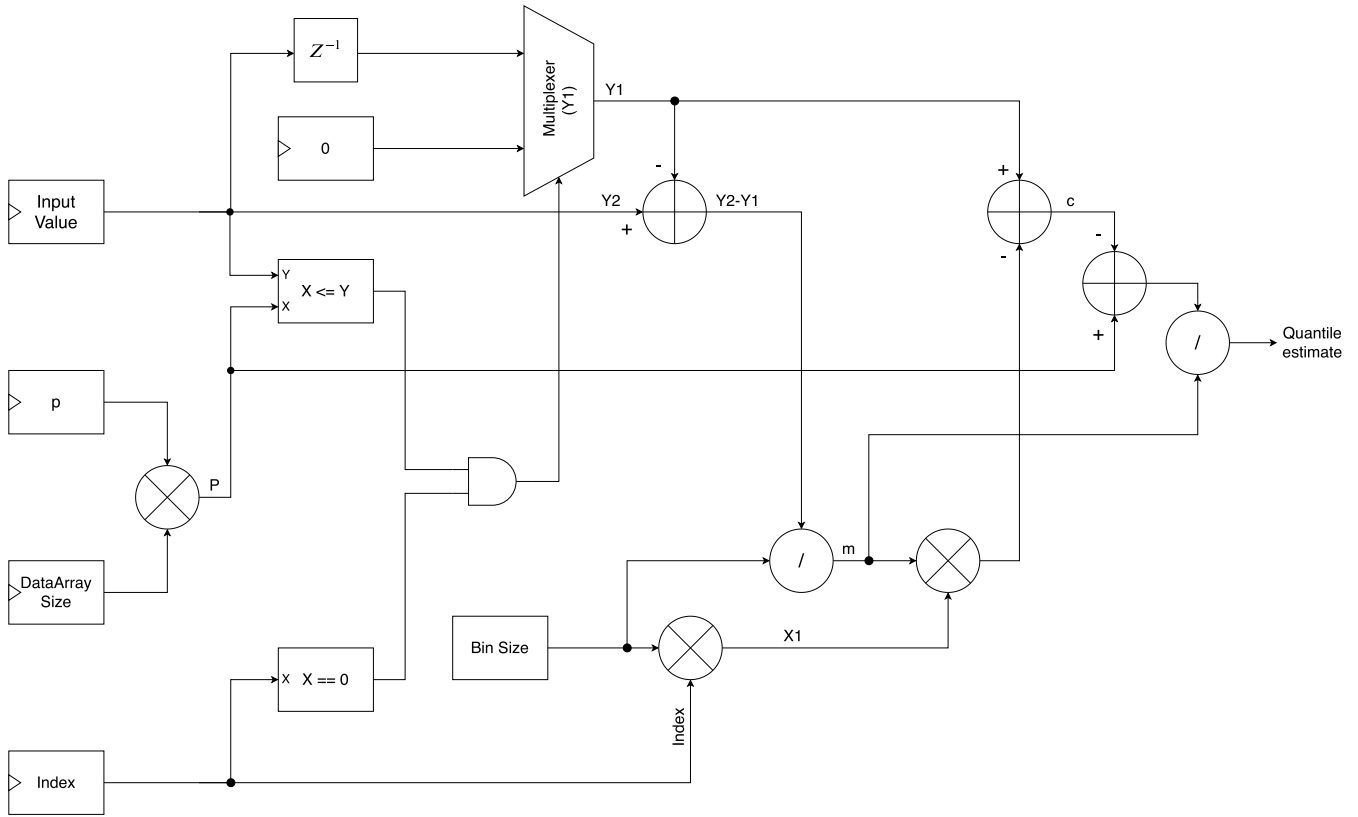


FIGURE 7. Logic design for the quantile estimation IP.

the CU data instead of the probability density, it is required to compute corresponding value  $P$  which is in the range of cumulative sum values.  $P$  is calculated by multiplying  $p$  with the data window size,  $N$  for which the histogram was calculated earlier.  $P$  is compared against each cumulative value to find the interval  $i$  where  $P$  is residing. A line is fitted to the two cumulative sum data points,  $\psi_i$  and  $\psi_{i+1}$  which are located to the left and right of  $P$ . The corresponding value,  $Q$  to  $P$  in the domain of the line is calculated by evaluating the inverse equation of the fitted line. The  $Q$  value thus calculated is given out as the quantile estimate for the corresponding probability value. The Algorithm 3 is implemented in *invcum* IP and the resulting logic design for this IP is given in Fig. 7. Its timing, latency information and resource utilization information are summarized in Table 1, Table 2 and Table 3 respectively.

**Algorithm 3** Quantile Estimation Algorithm

- 1: **Input:**  $C, p$
- 2:  $P \leftarrow N \times p$
- 3: Find the interval,  $(\psi_i \leq P < \psi_{i+1})$  where  $P$  resides in
- 4: Fit a line which goes through  $\psi_i$  and  $\psi_{i+1}$
- 5: Find the value  $Q$  corresponding to the value  $P$  in the domain of the fitted line using (10)
- 6: **Output:** Quantile estimate,  $Q$

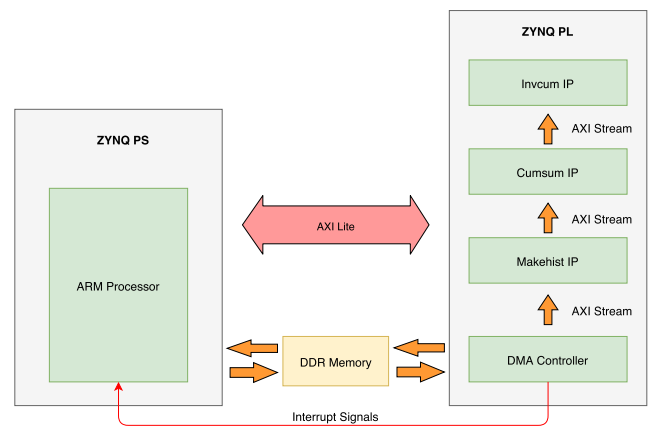


FIGURE 8. Complete system block diagram.

**C. IP INTEGRATION AND SYSTEM GENERATION**

Fig. 8 illustrates the complete system block diagram once the integration is done. It also shows the communication protocols which are used between different subsystems. Mainly, AXI protocol [34] is used for the control and configuration of the IPs and other modules which are residing in the PL. Xilinx adopted AXI protocol is called AXI4-Lite. AXI protocol is targeted for high performance, high frequency system designs and provides a high bandwidth and a low latency.



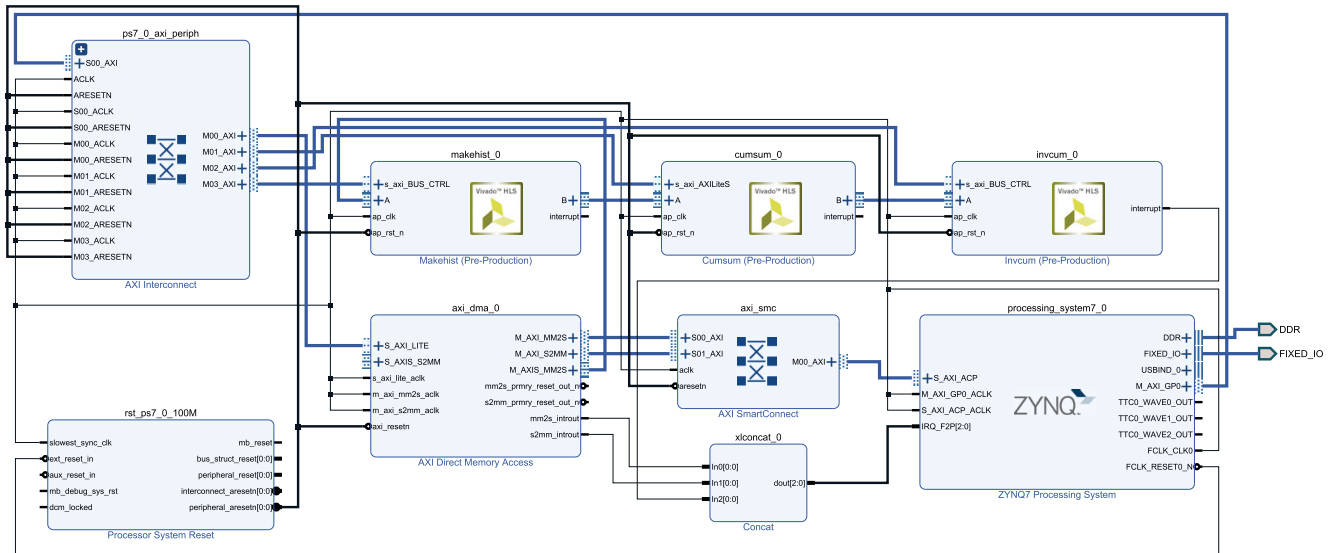


FIGURE 9. Integrated system in Xilinx Vivado.

We use *AXI4-Stream* protocol in conjunction with ACP of the ARM CPU in Zynq-7000 Ap SoC for high speed streaming data transfers. *AXI4-Stream* protocol is used in applications where the focus is on a data-centric and data-flow paradigm where the concept of address is not important [33]. *AXI4-Stream* behaves as a single unidirectional channel with a handshaking data flow. Due to this property, the mechanism to move data between IPs is efficient and fast. *AXI4-Stream* protocol is highly optimized for high performance data flow applications. Therefore, to fulfill low latency for our implemented design, the data transfer from the system memory (DDR memory) to the respective IP (*makehist* IP) is realized using the *AXI4-Stream* protocol. DMA controller with CPU interrupts takes care of the data movement from the system memory to the *makehist* IP independently with negligible amount of intervention from the ARM processor [35]. Due to this reason, the overhead on the processor on data transfer process is kept at minimal. Same protocol is used for the data transfers between the IPs also. Fig. 8 depicts the locations where *AXI4-Stream* protocol is used for data transfer operations. Once the data is streamed to the *makehist* IP, the data flows through all the three IPs sequentially as a stream. Each IP processes the data as per the algorithms described in section IV-B and the processed data is forwarded to the next IP in succession for further processing.

#### D. OVERALL DESIGN FLOW OF THE IMPLEMENTED SYSTEM

Fig. 9 shows block design implemented in Xilinx Vivado by integrating the IPs. It shows all the IPs with system interconnects and other peripheral IPs which are needed for the integration purposes. Interrupts of DMA controller are used to detect the completion of write data transaction between the system memory and the *makehist* IP. DMA controller is connected to the ACP port of the ARM processor.

Data streaming from system memory to the streaming device (*makehist* IP) takes place through the memory mapped to streaming (MM2S) channel. Data streaming from streaming device to the system memory takes place through the streaming to memory mapped (S2MM) channel. At the end of each transaction, DMA asserts an interrupt to notify the ARM processor, the completion of the data transaction. This information is used by the ARM processor to advance to the next state to read back the results and to schedule the next data transaction.

The system is synthesized and implemented in Xilinx Vivado after the block design. The bitstream for FPGA configuration is generated next. We export the implemented hardware in Xilinx Vivado and use Xilinx SDK for the bare-metal application development for the embedded ARM processor. Zynq-7000 AP SoC is configured through Xilinx SDK and the developed bare-metal application is executed in the ARM processor.

#### V. RESULTS AND DISCUSSION

In this section, we present the results related to the performance evaluation of the proposed quantile estimation algorithm. For testing the accuracy of the implemented algorithm, we use real block maximum CU values and compare with theoretical quantile estimates using the extreme value theory. CU data used for the evaluation purpose was actual data collected from the university's Wi-Fi network. More details on the method of collecting the CU data can be found in [27]. We evaluate the performance of the implemented algorithm by comparing the results with those from MATLAB. In our testing, ZedBoard was connected to a computer with MATLAB and the results were obtained by sending CU data to the ZedBoard and reading back the computed output from ZedBoard and comparing with MATLAB extreme value theory quantile estimates.

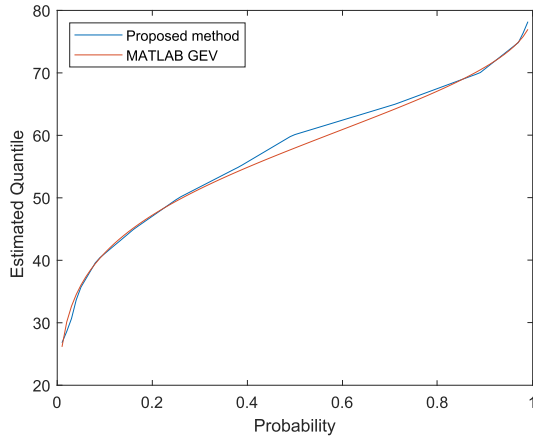


FIGURE 10. Estimated quantiles from the proposed method and MATLAB GEV method.

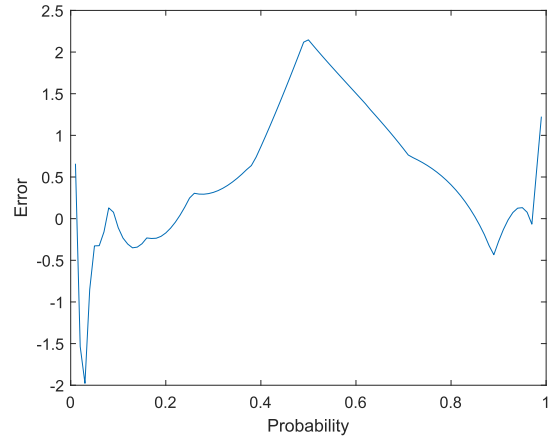


FIGURE 11. Error in the quantile estimates between the proposed method and MATLAB GEV method.

A. EVALUATION OF THE PROPOSED ALGORITHM

In this section, the performance of the proposed algorithm is evaluated against the GEV of MATLAB implementation. We use MATLAB GEV for comparison since it models the extreme values more accurately than the empirical method. The block maxima series of CU data used for the testing has been collected with a rate of 3 block maximum CU samples per minute for a period of 9 hours from university’s Wi-Fi network. Max values of CU data for the evaluation were partitioned to 1 hour data blocks on which the percentile estimates are made using the proposed algorithm and compared against the results from the MATLAB implementation of GEV.

For a data set of 1 hour, Fig. 10 shows the quantile estimates given by the proposed method and the GEV implementation of MATLAB for probabilities from 0.01 to 0.99. It is clear from the figure that throughout the probability range from 0.01 – 0.99, the estimated percentiles from the proposed method closely follow the percentile estimates calculated using GEV implementation of MATLAB. There are some deviations visible which are due to the approximation errors of histogram as compared to the exact pdf. This can be described as an effect of finite bin width of the histogram. Fig. 11 shows the error plot between the MATLAB GEV and the proposed method. We can observe that throughout the probability range, the variation of the error approximately resides in the interval [−2, 2.5]. The mean and the standard deviation of the error between the two methods are -0.469 and 0.891 respectively.

To evaluate the accuracy of quantile estimation of the proposed method, we plotted the estimated quantiles of the proposed method against the quantiles estimated by MATLAB GEV. Fig. 12 shows the resulting correlation plot between the two methods. We also fitted a regression model to the data set and the respective 95% confidence interval on the same figure. It is visible from the figure that most of the data points fall inside the 95% confidence interval. Therefore, with 95% of confidence, we can reason that the mean of the future observations would fall inside the confidence interval. This implies that the estimation performance of the proposed

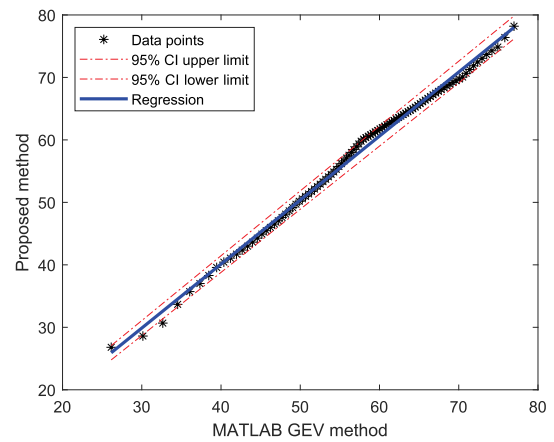
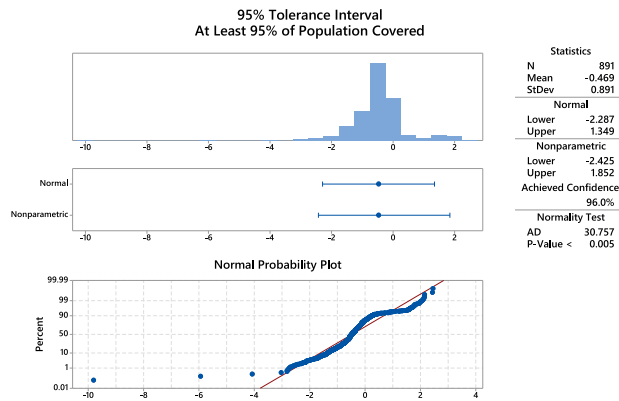


FIGURE 12. Correlation between the proposed method and MATLAB GEV method.

method would stay almost invariant through out random samples of the data set. The correlation coefficient between the estimated quantiles from the proposed method and MATLAB GEV is 0.995. Assuming a significance level (denoted as  $\alpha$ ) of 0.005, we receive a probability value ( $p$ -value) equal to 0.000 which tells us whether the correlation coefficient is significantly different from 0. With  $p$ -value  $\leq \alpha$ , we can presume that the calculated correlation coefficient is significant. Therefore, we can conclude that the performance of the proposed method is closely related to the original GEV method.

To study the behavior of error in quantile estimation, we calculated the difference between the results from the proposed method and the MATLAB GEV method. Using the statistical software Minitab, we obtained the tolerance interval plot. Tolerance interval is an important measure which gives the range that is likely to accommodate a specified proportion of the population. Confidence level for the tolerance interval gives the likelihood that the interval would cover the specified proportion. Therefore, we can use the tolerance interval for the error to predict the future values of error with a specified confidence level.



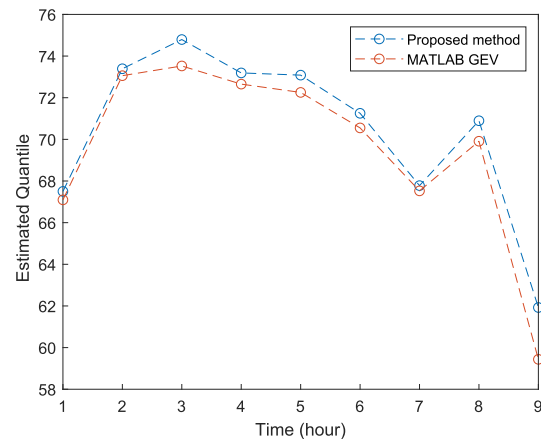
**FIGURE 13.** Tolerance interval plot for the error between the *MATLAB* GEV and the proposed method.

Fig. 13 depicts the tolerance interval plot for the calculated error between the proposed method and *MATLAB* GEV method. It also shows the normality test for the error. The calculated  $p$ -value for the normality test is lower than 0.005. Therefore, we should reject the null hypothesis and come to the conclusion that the error does not follow a normal distribution. Consequently, we can use the non-parametric tolerance interval for the error. For our specific data set, the lower and upper bounds of the tolerance interval for the error are given as  $-2.425$  and  $1.852$  respectively. Therefore, with a confidence level of 95%, we can expect that future errors which would be generated from the proposed method would fall inside this tolerance interval.

Fig. 14 shows the estimated quantiles of the proposed method and *MATLAB* GEV over time for a probability of 0.8. It is clear from the figure that the proposed method closely follows the behavior of *MATLAB* GEV across different samples. There are some deviations in the estimated values of the proposed method from *MATLAB* GEV. But, according to earlier observations in the behavior of the error, we know with a confidence level of 95% that the error would reside in the tolerance interval,  $[-2.425, 1.852]$ . Therefore, it can be concluded that the accuracy of the results of the proposed method is higher at a respectable level.

Despite the simplicity, our evaluation of the performance of the proposed algorithm gives positive results. The quantile estimates given by the proposed algorithm have very high accuracy and have very little deviations from the estimates of actual GEV. With the performance gain and reduced implementation complexity, small deviations from the expected results are tolerable. Therefore, with a considerable confidence, we can rely on the quantile estimates generated by the proposed method to apply for different resource utilization scenarios of wireless communication channels.

Our measurement results have shown that when 50th quantile contains 70% or above CU, then the channel can be considered congested as the quality of service of real time applications like Skype Video and streaming services degrade significantly. By forecasting a particular quantile, we can estimate whether the channel is experiencing congestion



**FIGURE 14.** Quantile estimation for  $p = 0.8$  across time for proposed method and *MATLAB* GEV method.

or not. This information can be utilized by a cloud controller in making proactive decisions on resource allocation in a wireless network. Hence, this ensures that the required service level of a wireless network is satisfied.

## VI. CONCLUSION

Recently, it has been recognized that real-time data analytics play an important role in achieving efficient network control, configuration and management. Real-time radio frequency data analytics require that hundreds of millions of streaming samples to be processed within a second and therefore, hardware acceleration using FPGAs can be considered more appropriate. In this paper, we have proposed a real-time quantile-based resource utilization estimator module for wireless networks using an FPGA. The proposed method has less complexity and can perform the data analytics in real time.

As a proof of concept, we have presented our solution with respect to estimating quantiles of real frequency CU data. Due to the bursty nature of CU data, we used the block maxima series of CU data which can be modeled using the GEV theory. The proposed method is implemented using *Xilinx Zynq-7000* series AP SoC board using *Vivado*, *Vivado HLS* and *Xilinx SDK* along with *MATLAB*. We thoroughly evaluated the performance and accuracy of the proposed method against the results obtained from the theoretical method using GEV theory tool in *MATLAB*. The comparison of the results reveals that the proposed algorithm performs almost equal to the theoretical implementation of GEV and the results are within a very small margin of error. The proposed method can be used in streaming data and can be used with high throughput applications requiring very low latency. Therefore, the implemented device can easily be used to perform quantile estimation and can be utilized for forecasting congestion in wireless frequency spectrum utilization.

## ACKNOWLEDGMENT

The authors would like to thank the Xilinx University Program (XUP) for providing them a donation of

four ZedBoards, which were used in this article for implementing the proposed forecasting device.

## REFERENCES

- [1] 5G Americas. *New Services & Applications With 5G Ultra-Reliable Low Latency Communications*. [Online]. Available: [https://www.5gamericas.org/wp-content/uploads/2019/07/5G/5G\\_Americas\\_URLLC\\_White\\_Paper\\_Final\\_updateJW.pdf](https://www.5gamericas.org/wp-content/uploads/2019/07/5G/5G_Americas_URLLC_White_Paper_Final_updateJW.pdf)
- [2] L.-C. Wang and S.-H. Cheng, "Data-driven resource management for ultra-dense small cells: An affinity propagation clustering approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 267–279, Jul. 2019.
- [3] Cisco meraki best practice design. [Online]. Available: [https://documentation.meraki.com/Architectures\\_and\\_Best\\_Practices/Cisco\\_Meraki\\_Best\\_Practice\\_Design](https://documentation.meraki.com/Architectures_and_Best_Practices/Cisco_Meraki_Best_Practice_Design)
- [4] S. Ludwig et al., "A 5G architecture for the factory of the future," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, Sep. 2018, pp. 1409–1416.
- [5] T. Cooklev, J. Darabi, C. Mcintosh, and M. Mosaheb, "A cloud-based approach to spectrum monitoring," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 2, pp. 33–37, Apr. 2015.
- [6] F. Rodrigues and F. C. Pereira, "Beyond expectation: Deep joint mean and quantile regression for spatio-temporal problems," 2018, *arXiv:1808.08798*. [Online]. Available: <http://arxiv.org/abs/1808.08798>
- [7] F. Restuccia and T. Melodia, "Big data goes small: Real-time spectrum-driven embedded wireless networking through deep learning in the RF loop," 2019, *arXiv:1903.05460*. [Online]. Available: <http://arxiv.org/abs/1903.05460>
- [8] Oracle. (Apr. 2015). *Improving Communications Service Provider Performance with Big Data: Architect's Guide and Reference Architecture Introduction*. Accessed: Jun. 7, 2019. [Online]. Available: <http://www.oracle.com/us/technologies/big-data/big-data-communications-2528130.pdf>
- [9] Riverbed. (Jun. 2019). *Riverbed OPNET NetOne*. Accessed: Jun. 4, 2019. [Online]. Available: [https://www.riverbed.com/document/fpo/media-cms/Riverbed\\_OPNET\\_NetOne.pdf](https://www.riverbed.com/document/fpo/media-cms/Riverbed_OPNET_NetOne.pdf)
- [10] Cisco. Feb. 2015. *MATE Design User Guide*. Accessed: Jun. 4, 2019. [Online]. Available: [https://www.cisco.com/c/dam/en/us/td/docs/net\\_mgmt/wae/6-1/design/user/guide/MATE\\_Design\\_User\\_Guide.pdf](https://www.cisco.com/c/dam/en/us/td/docs/net_mgmt/wae/6-1/design/user/guide/MATE_Design_User_Guide.pdf)
- [11] P. P. Mariño. (Jul. 2018). *Net2Plan 0.6.0 User's Manual*. Accessed: Jun. 5, 2019. [Online]. Available: <http://www.net2plan.com/documentation/current/help/usersGuide.pdf>
- [12] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, "Big data empowered self organized networks," in *Proc. 20th Eur. Wireless Conf.*, May 2014, pp. 1–8.
- [13] M. S. Hadi, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Big data analytics for wireless and wired network design: A survey," *Comput. Netw.*, vol. 132, pp. 180–199, Feb. 2018.
- [14] S. Barmounakis, A. Kaloxylas, P. Spapis, P. Magdalinos, N. Alonistioti, and C. Zhou, "Data analytics for 5G networks: A complete framework for network access selection and traffic steering," *Int. J. Adv. Telecommun.*, vol. 11, nos. 3–4, pp. 101–114, 2018. [Online]. Available: [http://www.iariajournals.org/telecommunications/tele\\_v11\\_n34\\_2018\\_paged.pdf](http://www.iariajournals.org/telecommunications/tele_v11_n34_2018_paged.pdf)
- [15] 5G: *System Architecture for the 5G System*, document TS 23.501 version 15.2.0 Release 15. 3GPP, Jun. 2018. Accessed: Jun. 8, 2019. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_ts/123500\\_123599/123501/15.02.00\\_60/ts\\_123501v150200p.pdf](https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf)
- [16] A. Kaloxylas, "Application of data mining in the 5G network architecture," in *13th Int. Conf. Digit. Telecommun. (ICDT)*, Aug. 2018, pp. 39–44.
- [17] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1666–1677, Aug. 2015.
- [18] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2017, pp. 1067–1075.
- [19] J. H. B. Deane, G. G. Johnstone, A. W. Ledford, and M. J. Underhill, "Extreme value theory applied to multi-channel communication systems," *Electron. Lett.*, vol. 33, no. 10, pp. 832–833, 1997.
- [20] S. Wei, D. L. Goeckel, and P. E. Kelly, "A modern extreme value theory approach to calculating the distribution of the peak-to-average power ratio in OFDM systems," in *Proc. IEEE Int. Conf. Commun. Conf. (ICC)*, vol. 3, May 2002, pp. 1686–1690.
- [21] P. Radojkovic, P. M. Carpenter, M. Moreto, V. Cakarevic, J. Verdu, A. Pajuelo, F. J. Cazorra, M. Nemirovsky, and M. Valero, "Thread assignment in Multicore/Multithreaded processors: A statistical approach," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 256–269, Jan. 2016.
- [22] Intel. (Jun. 2019). *FPGA Inline Acceleration for Streaming Analytics. Intel Whitepaper*. Accessed: Jun. 11, 2019. [Online]. Available: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01278-fpga-inline-acceleration-for-streaming-analytics.pdf>
- [23] Bertin. (2016, May) *GPU vs FPGA Performance Comparison*. Bertin. Accessed: Jun. 15, 2019. [Online]. Available: [http://www.bertensp.com/pdf/whitepaper/BWP001\\_GPU\\_vs\\_FPGA\\_Performance\\_Comparison\\_v1.0.pdf](http://www.bertensp.com/pdf/whitepaper/BWP001_GPU_vs_FPGA_Performance_Comparison_v1.0.pdf)
- [24] S. A. Fahmy, "Histogram-based probability density function estimation on FPGAs," in *Proc. Int. Conf. Field-Program. Technol.*, Dec. 2010, pp. 449–453.
- [25] A. Gramacki, M. Sawyerain, and J. Gramacki, "FPGA-based bandwidth selection for kernel density estimation using high level synthesis approach," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 64, no. 4, pp. 821–829, Dec. 2016.
- [26] C. A. Ganewattha, "Wireless channel load stress analysis using FPGAs at the edge," M.S. thesis, Centre Wireless Commun., Univ. Oulu, Oulu, Finland, 2019. [Online]. Available: <http://jultika.oulu.fi/Record/nbnfioulu-201910162972>
- [27] Z. Khan, J. J. Lehtomaki, E. Hossain, M. Latva-Aho, and A. Marshall, "An FPGA-based implementation of a multifunction environment sensing device for shared access with rotating radars," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 11, pp. 2561–2578, Nov. 2018.
- [28] S. Coles, *An Introduction to Statistical Modeling of Extreme Values* (Springer Series in Statistics), 1st ed. London, U.K.: Springer-Verlag, 2001.
- [29] AVNET. (Jan. 2014). *ZedBoard: (Zynq Evaluation and Development) Hardware User's Guide*. Accessed: Jun. 8, 2019, [Online]. Available: [http://zedboard.org/sites/default/files/documentations/ZedBoard\\_HW\\_UG\\_v2\\_2.pdf](http://zedboard.org/sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf)
- [30] Xilinx. (Dec. 2018). *UG902 (v2018.3): Vivado Design Suite User Guide: High-Level Synthesis*. Accessed: Jun. 8, 2019, [Online]. Available: [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2018\\_3/ug902-vivado-high-level-synthesis.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug902-vivado-high-level-synthesis.pdf)
- [31] Programmable SoC Software Developers Guide. Xilinx. Accessed: Jun. 8, 2019. [Online]. Available: [https://www.xilinx.com/support/documentation/user\\_guides/ug821-zynq-7000-swdev.pdf](https://www.xilinx.com/support/documentation/user_guides/ug821-zynq-7000-swdev.pdf)
- [32] Xilinx. (Jul. 2018). *DS190 (v1.11.1): Zynq-7000 SoC data sheet: Overview*. Accessed: Jun. 8, 2019, [Online]. Available: [https://www.xilinx.com/support/documentation/data\\_sheets/ds190-Zynq-7000-Overview.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf)
- [33] Xilinx. (Mar. 2011). *UG761 (v13.1): AXI Reference Guide*. Accessed: Jun. 8, 2019. [Online]. Available: [https://www.xilinx.com/support/documentation/ip\\_documentation/ug761\\_axi\\_reference\\_guide.pdf](https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf)
- [34] ARM. (Mar. 2018). *AMBA AXI Protocol V1.0. ARM*. Accessed: Jun. 8, 2019, [Online]. Available: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ih0022b/index.html>
- [35] Xilinx. (Apr. 2018). *PG021: LogiCORE IP Product Guide*. Accessed: Jun. 8, 2019. [Online]. Available: [https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_dma/v7\\_1/pg021\\_axi\\_dma.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf)



**CHANAKA GANEWATTHA** received the B.Sc. degree in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2016, and the M.Sc. degree in electrical engineering from the University of Oulu, Finland, in 2019, where he is currently pursuing the Ph.D. degree in communications engineering.

From 2016 to 2018, he was an Electronic Engineer with Paracom Technologies, Sri Lanka. His research interests include FPGA-based digital hardware implementations, software-defined radio, algorithms in signal processing, and machine learning.



**ZAHEER KHAN** received the M.Sc. degree in electrical engineering from University College Borås, Sweden, in 2007, and the Dr.Sc. degree in electrical engineering from the University of Oulu, Finland, in 2011. He was a Research Fellow/Principal Investigator with the University of Oulu, from 2011 to 2016, where he is currently an Adjunct Professor. He was a Tenured-Track Lecturer with the University of Liverpool, U.K., from 2016 to 2017. His research interests include

the implementation of advanced signal processing and wireless communications algorithms on Xilinx FPGAs and Zynq system-on-chip (SoC) boards, application of game theory to model distributed wireless networks, prototyping access protocols for wireless networks, the IoT location tracking systems, cognitive and cooperative communications, and wireless signal design. He was a recipient of the Marie Curie Fellowship, from 2007 to 2008.



**JANNE J. LEHTOMÄKI** received the Ph.D. degree from the University of Oulu, Finland, in 2005. He is currently an Adjunct Professor with the Centre for Wireless Communications, University of Oulu. He spent the Fall 2013 semester at the Georgia Tech, Atlanta, USA, as a Visiting Scholar. His current research interests include spectrum measurements and terahertz band wireless communication. He has coauthored the article, which received the Best Paper Award at the IEEE WCNC

2012. He was the General Co-Chair of the IEEE WCNC 2017 International Workshop on Smart Spectrum, the TPC Co-Chair of the IEEE WCNC 2015 and the 2016 International Workshop on Smart Spectrum, and the Publicity/Publications Co-Chair of the ACM NANOCOM, from 2015 to 2017. He served as a Guest Associate Editor for the *IEICE Transactions on Communications Special Section*, from February 2014 to July 2017, and a Managing Guest Editor for *Nano Communication Networks Special Issue*, in June 2016. He is also an Editorial Board Member of *Physical Communication*.



**MARJA MATINMIKKO-BLUE** received the Dr.Sc. (Tech) and Ph.D. degrees. She is currently a Senior Research Fellow and an Adjunct Professor in spectrum management with the Centre for Wireless Communications (CWC), University of Oulu, Finland. She conducts interdisciplinary research on future mobile communication networks from business, technology, and regulatory perspectives. She is also a Research Coordinator of the 6G Flagship (6G-Enabled Wireless Smart Ecosystem

and Society-6Genesis Flagship). She has published over 120 scientific articles and participated in spectrum regulatory forums in Europe (CEPT) and globally (ITU), including preparation of over 100 contributions and chairmanship of cognitive radio system studies. Her current research interests include mobile communication system design, spectrum sharing techniques, new spectrum valuation and licensing models, and local operator models for 5G networks.

...