IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 24, 2021, accepted October 12, 2021, date of publication October 15, 2021, date of current version October 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120668

A Lightweight and Real-Time Hardware Architecture for Interference Detection and Mitigation of Time Synchronization Attacks Based on MLP Neural Networks

NILOOFAR OROUJI[1], **MOHAMMAD REZA MOSAVI**[2], **AND DIEGO MARTÍN**[1]

[1]ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]Department of Electrical Engineering, Iran University of Science and Technology, Narmak, Tehran 13114-16846, Iran

Corresponding author: Diego Martín (diego.martin.de.andres@upm.es)

ABSTRACT Stationary GPS receivers provide time information for critical infrastructures, such as phasor measurement units (PMUs), communication networks, and financial systems. Therefore, they are prone to a specific type of spoofing attack called time synchronization attack (TSA), which affects time information such as clock offset and clock drift. The receiver's position remains constant during the attack; hence, attack detection and mitigation are challenging. Various countermeasures have been suggested to mitigate TSA effects. However, they are mainly software-based and are exploited to protect software implemented software-defined radios (SDRs). In this research, two hardware protection approaches are contributed for hardware-based SDRs based on multi-layer perceptron neural network (MLP NN) with sigmoid activation function. The most challenging part of MPL NN implementation is the activation function approximation. Therefore, two lightweight architectures are proposed for sigmoid function implementation. Linear approximation and look-up table (LA-LUT) and piece-wise linear approximation (PLA) are exploited for this task. The synthesis results demonstrate that the PLA approach has a slightly higher resource utilization in comparison to LA-LUT, while this method is more accurate. The mean squared error (MSE) of the PLA approach is equal to 0.019, which is 57% better than the LA-LUT approach with an MSE of 0.033. Furthermore, the designs are evaluated by two conventional types of TSA. According to the results, both methods are lightweight, and they only consume less than 0.3% of slice registers, 5% of slice LUTs, and 8% of DSP48E1Ss. Furthermore, they are real-time, and can mitigate the attack consequences; however, the PLA architecture has a better performance compared to LA-LUT.

INDEX TERMS Hardware implementation, FPGA, approximation of sigmoid function, spoofing attacks.

I. INTRODUCTION

Nowadays, many crucial infrastructures, such as power grids [1], communication towers, and financial systems, depend on global positioning system (GPS) for acknowledging the accurate time [2]. GPS signals are weak at the earth's surface, and the signal structure is known publicly. Furthermore, civil GPS signals have no protection and correction mechanisms; therefore, an experienced adversary can easily alter the timing information of the target receivers [3].

The receivers in the mentioned infrastructures are stationary; thus, the adversary can place a receiver-spoofer device near the target receiver, transmitting a manipulated version of the signal with a slightly higher power [4]. The receiver-spoofer device extracts the code phase and Doppler frequency of the genuine signal; therefore, the resultant spoofing signal is very similar to the authentic one. This type of attack is known as time synchronization attack (TSA) and is considered an intermediate spoofing attack. Generally, TSA only manipulates the time information while the receiver position remains constant [5]. Due to the hidden nature of TSA and the similarity between the spoofing signal and the

The associate editor coordinating the review of this manuscript and approving it for publication was Easter Selvan Suviseshamuthu.

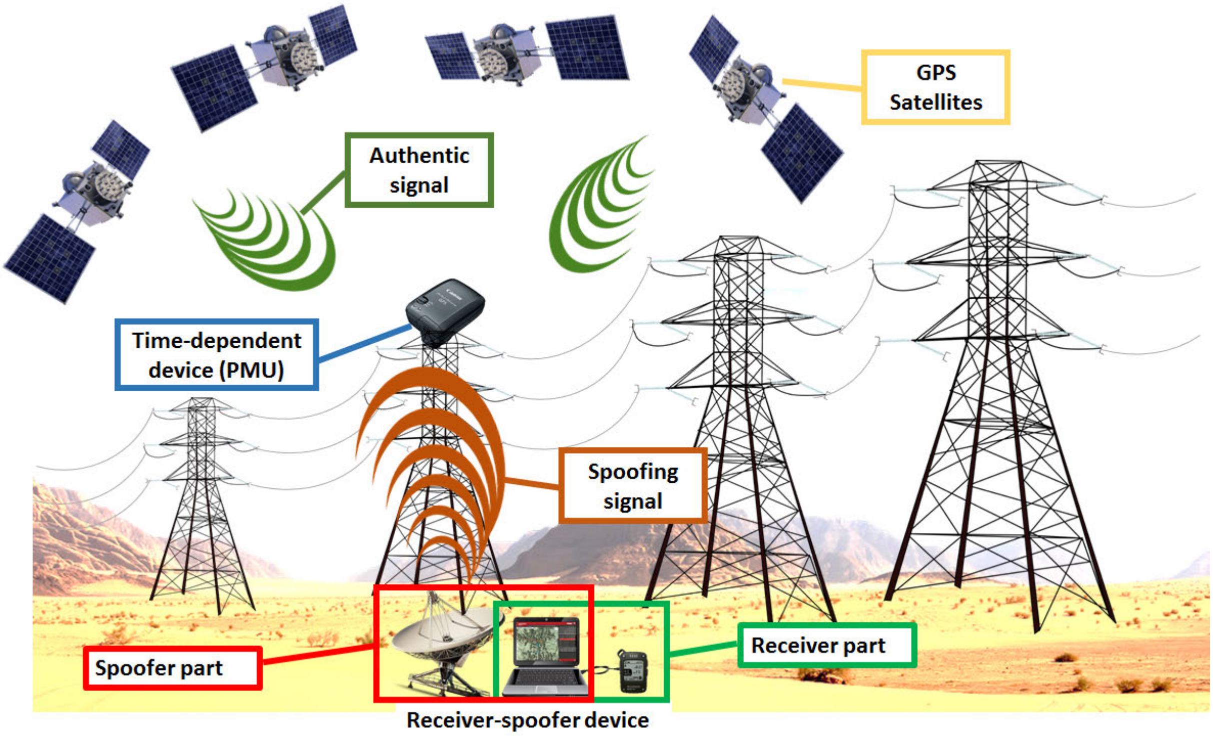

FIGURE 1. Scheme of TSA on a PMU of smart grids.

genuine one, almost all commercial GPS receivers can be spoofed easily [6].

A. ATTACK SCHEME

A commercial GPS receiver utilizes the GPS signals to obtain its position, velocity, and time, which is generally known as PVT solution.

In stationary receivers, the position is constant, and velocity is zero; therefore, only the time information concerns the application. The GPS receivers exploit low-cost crystal oscillators to maintain the time, which cannot provide high accuracy time information. Regarding the clock oscillators behavior [7], the receiver clock t_u has an offset in comparison to the GPS clock t^{GPS} [8]:

$$t_u = t^{GPS} + dt_u, \tag{1}$$

where dt_u is the receiver clock offset. The erroneous clock offset leads to incorrect receiver time. In critical infrastructures such as phasor measurement units (PMUs) and smart grids, accurate time information is employed for attaching timestamps to the voltage and current measurements of the network. Inaccurate timestamps can cause false alarms or misdeclaration of critical situations in the network, which results in disastrous and catastrophic incidents [9].

In the attack scheme, as shown in Fig. 1, the adversary receives the genuine signals of the in-sight satellites and alters the clock offset information of the receiver by modifying the pseudo ranges of the satellites [5]. It can transmit the spoofing signal to the target receiver utilizing the spoofer part of the device. The higher power of the spoofing signal and its similarity to the authentic one convince the receiver to track the spoofed replica as the genuine signal. In this attack, manipulating the pseudo ranges of all satellites results in the unchanged receiver position and modified clock offset, which leads to erroneous timestamps [6].

B. COUNTERMEASURES

As experienced spoofers seek new ways to create novel spoofing attacks, researchers contribute countermeasures to repulse the attack and mitigate its effects. The contributed researches can be classified into four main categories [10]: signal processing approaches [6], [11], [12], multi-antenna receivers [13]–[16], validations with other GNSS signals [17], [18], and cryptographic techniques [19], [20].

The cryptographic techniques require a structural update of the GPS signal; thus, it has not been operated practically on civil signals. Furthermore, the multi-antenna receivers and GNSS validations entail more pieces of equipment and increase the defense and protection costs. Signal processing is the most popular method due to its ease of use and minimum requirements, and a firmware update can convey the newest protection techniques to the receiver. This method suggests a dynamic solution to encounter the evolving nature of TSA by offering firmware updates.

The GPS receivers generally include hardware components, which are not adaptable and updatable. In contrast, upgradability is one of the main features of the software-defined radios (SDRs); thus, the signal processing techniques are mainly implemented on SDRs [21]. However, the SDR approach cannot compete with the computational power of hardware receivers [22], and often their solutions are not real-time. Consequently, SDRs utilize programmable hardware resources to increase their computational resources and maintain upgradability.

A novel classification of GNSS SDRs has been presented in [21] to classify SDRs based on their implementation approaches in four categories. The first category includes the prototyping softwares (P-SWs), where the baseband signals of the GNSS RF part are passed through the SW-implemented blocks: acquisition, tracking, and navigation solution. This prototyping approach is implemented in high-level programming applications, such a Matlab [8], [23] or Matlab and Simulink [24], LabVIEW [25], [26], and open-source tools [27]. Although this prototyping approach is fast and easy to exploit, it is not real-time most of the time [28], [29].

In the second category, known as the host PC method, the high-level programming framework is omitted, and the host PC mainly executes acquisition, tracking, and navigation blocks. Linux OS [30], graphical user interfaces (GUIs) [31], application programming interfaces (APIs) [32], GPU libraries [33], and C++ based software [34] are among the popular SDRs in this category. The host PC solutions can be real-time, non-real-time, or both. Note that elimination of high-level applications leads to less flexibility and exploitation ease.

DSPs and embedded general purpose processors (GPPs) are included in the third category, along with a host PC to exhibit the results. The methods in this category utilize DSP accelerators [35] and bit-wise operations [36] to expedite the SDR procedure. An example of GPP exploitation for LSTM-based GNSS spoofing detection is also presented in [37]. DSP boards are the main realization tools in this category; thus, the implemented SDR is generally real-time.

The last category includes the FPGA-based implementation, which has the least flexibility and utilization ease in the classification of [21]. In this approach, the required correlations are implemented in the FPGA platform, and the host PC determines the hardware configurations. Although FPGA implementations of SDR are the least flexible ones, the upgradability nature of SDR is still preserved. Therefore, even this type of SDR can receive firmware updates to protect the receiver against potential malicious attacks. Furthermore, these approaches can be executed in real-time [38], non-real-time [39], or both ways [40].

C. PAPER CONTRIBUTION

Although various types of SDRs have been proposed in the last few years, a few pieces of research have focused on the TSA mitigation techniques to develop a resilient receiver. Any protection against TSA in critical infrastructures should be real-time to provide high accuracy for the application. Protection procedures require dedicated computational resources to operate in real-time; hence, the P-SW and host PC approaches are not suitable for these implementations.

In this research, an open-loop FPGA-based architecture of [41] is proposed to assist in filling this research gap. The suggested method in [41] is a TSA detection and mitigation approach utilizing an multi-layer perceptron neural network (MLP NN), which is classified in signal processing techniques and P-SW implementations. According to the reported results, the method has been outperformed the extended Kalman filter [42], Luenberger observer [43], and robust estimator[5]; therefore, it is a proper candidate for hardware implementation.

The contributions of this research can be listed as follows:

- The proposed algorithm in [41] is modified to reduce its sensitivity to approximation errors under attack conditions.
- The defense algorithm can detect and mitigate TSAs which affect the clock offset and drift information.
- Two high-precision approximations of sigmoid function are presented to provide the highest possible accuracy along with the least resource utilization.
- The proposed hardware architecture has been designed as an open-loop extension; therefore, any compatible and configurable hardware SDR can employ its protection features.
- The proposed designs are lightweight; therefore, they can be implemented on the FPGA-based SDRs easily.
- The design exploits integers instead of floating-point numbers; hence, the computational overhead is decreased drastically.
- The concurrent feature of FPGA implementation accelerates the protection algorithm execution. Therefore, the proposed architecture can be exploited in real-time, non-real-time, or both ways.

This research is organized as follows: Section II surveys the sigmoid approximation methods and introduces the proposed approximations. Section III includes the proposed hardware architecture and design considerations, while section IV discusses overall performance evaluations and resource utilization. Furthermore, section IV assesses the performance of the architecture under the conditions of two TSA attacks. Eventually, section V concludes the research.

D. SIGMOID APPROXIMATION

Real-time execution and accuracy are among the main features of any protection algorithm. While the accuracy mainly depends on the innovations of the algorithm, the execution speed relies on the implementation platform. In neural networks (NNs), precision emerges from its design configurations, such as the number of layers and neurons, type of the activation function, and selected input features. The proper choice of implementation platforms based on the application has a significant impact on the execution time. Since the NN structure provides the parallel implementation opportunity, a hardware implementation platform accelerates the execution time.

So far, the hardware implementation has a significant advantage in comparison to software-based approaches; however, there is a considerable problem in hardware implementation of NNs: activation function approximation. Generally, NNs utilize sigmoid and hyperbolic tangent functions as activation functions. The direct implementation of these functions consumes a massive amount of hardware resources, and for deep (or large) NNs, it is almost impossible. Many pieces of research have been proposed various solutions to overcome this problem. The exploitation of look-up tables (LUTs) is the first and the most feasible solution. In this approach, a table of activation function samples is stored in memory, and a mapping mechanism maps each input to a specific row of the table. A lightweight LUT-based approximation has been proposed in [42], which is fast and does not consume valuable computational resources. However, its accuracy depends on the number of samples in the table.

As an alternative solution, coordinate rotational digital computer (CORDIC) algorithm, available in Xilinx LogiCORE IP core, provides an opportunity for direct implementation of sigmoid and hyperbolic tangent functions [44]. Although this IP core can implement trigonometric and hyperbolic functions, it consumes a considerable amount of computational resources.

Various mathematical approximations have been employed to increase accuracy along with reasonable usage of hardware resources. In [45], a controlled approximation method based on the Taylor theorem is presented that bounds the approximation error by its Lagrange form. A high accuracy approximation method for sigmoid and hyperbolic tangent is contributed in [46], which utilizes the McLaurin series interpolation and Pade approximation, and its reconfigurable implementation is presented in [47]. Second-order and piece-wise linear approximations (PLA) are also amongst popular solutions to approximate sigmoid function [48].

Exploiting floating-point numbers in hardware implementations leads to high computational resource utilization. Therefore, in the following subsections, two efficient approaches for fixed-point sigmoid implementation are proposed. The first proposed method is based on LUT and linear approximation combination (LA-LUT), while the second approach focuses on the PLA approach.

E. PROPOSED LA-LUT APPROACH

An efficient binary representation of the sigmoid function is proposed in [49]. However, the exploited sigmoid function in [41] has slight differences from the one presented in [49]. Since the first goal of this research is the efficient implementation of the proposed MPL NN of [41], the sigmoid function is modified to the binary version for further comparisons and sampling of LUT and linear approximation combination (LA-LUT).

Consider the sigmoid function, exploited in [41], as (2):

$$\sigma\left(x\right)=\frac{2}{1+e^{-2x}}-1. \tag{2}$$

The exponent base e is replaced with *2* to represent binary numbers, variable x is substituted with n to demonstrate integers, and e^{-2x} is replaced by $2^{-\left(\frac{2n}{\log(2)}\right)}$, as suggested in [49]:

$$\sigma\left(n\right)=\frac{2}{1+2^{-\left(\frac{2n}{\log(2)}\right)}}-1, \tag{3}$$

As mentioned in [49], a proper scaling factor 2^k results in a function with integer outputs. Inspired by [48], $k = \mathit{10}$ is selected to scale the function:

$$\sigma\left(n\right)=\frac{2\times 2^{10}}{1+2^{-\left(\frac{2n}{\log\left(2\right)\times 2^{10}}\right)}}-1\times 2^{10}. \tag{4}$$

The effective input range of the sigmoid function represented in (2) is *(−8, 8)*, and the output is in the range of *(−1,1)*. After scaling, as shown in (4), the inputs are in the range of *(−8192, 8192)*, and the function output range is *(−1024, 1024)*. It should be noted that the inputs between two integers are rounded to the higher value, which causes a slight degradation in the approximation precision. This issue will be discussed in section IV.

The binary scaled version of the sigmoid function presented in (4) is the basis of the proposed LA-LUT approach. Furthermore, the sigmoid function of (4) is symmetrical:

$$\sigma\left(-n\right)=-\sigma\left(n\right). \tag{5}$$

Therefore, sampling the positive half is adequate to constitute the LA-LUT. For $n < x''FA''$, the function can be approximated by a simple line:

$$\sigma_{est}\left(n\right)=n, \quad forn < x''FA'', \tag{6}$$

For $n > x''FA''$, the function has been sampled to fulfill (7) based on the required precision for PMU application:

$$e_{est}\left(n\right)<5. \tag{7}$$

In which, the estimation error $e_{est}\left(n\right)$ is defined as:

$$e_{est}\left(n\right)=\sigma\left(n\right)-\sigma_{est}\left(n\right), \tag{8}$$

where n is an integer as function input, $\sigma\left(n\right)$ is the output of (4), and $\sigma_{est}\left(n\right)$ is the approximation result. In addition, the corresponding derivative error can be expressed as:

$$e_{dest}\left(n\right)=\frac{d\sigma\left(n\right)}{dn}-\frac{d\sigma_{est}\left(n\right)}{dn}. \tag{9}$$

The estimation error threshold has been selected to achieve an optimum balance between precision and resource usage, and it is obtained through a set of tests. Eventually, 110 samples of (4) fulfill (7). LA-LUT estimation, estimation errors, and corresponding derivative errors are exhibited in Fig. 2, and the mean squared error (MSE) of the LA-LUT approach is equal to 0.033.

F. PROPOSED PLA APPROACH

In the piece-wise linear approximation (PLA), ten lines have been exploited for sigmoid approximation, as shown in Table 1. Furthermore, the approximation results, such as comparison to the main function, estimation errors, and derivative errors, are demonstrated in Fig. 3.

LA-LUT and PLA both have fulfilled the condition of (7), regarding Fig. 2 and Fig. 3. However, PLA has a lower e_{est} (n) and e_{dest} (n) in comparison to LA-LUT. The MSE of the PLA approach is equal to 0.019, which is 57% better than the LA-LUT approach. Generally, LUT-based approaches are known for their fair resource usage in comparison to other methods. Although, a LUT with higher rows requires a more sophisticated mapping mechanism, which might affect the overall resource usage. In the next section, two hardware architectures for each method are proposed, which focus on reducing resource usage and increasing precision.

II. PROPOSED OPEN-LOOP ARCHITECTURE

In this section, an open-loop architecture for a three-layer MLP NN based on [41], is proposed and discussed in detail. The design aspects of MLP NN are also available in [41]. Furthermore, two sigmoid hardware realizations are presented regarding approximations of section II. In the third sub-section, the detection and mitigation algorithm of [41] is modified to tolerate the inaccuracies caused by the hardware implementation and approximations.

TABLE 1. Proposed Piece-wise Linear Approximation (PLA). All numbers are presented in hexadecimal form.

Part	Linear approximation	Range
1	$x"1" * n + x"3B8"$	$x”0” < n < x”12C”$
2	$x"2" * n + x"371"$	$x”12C” < n < x”258”$
3	$x"3" * n + x"32E"$	$x”258” < n < x”352”$
4	$x"5" * n + x"2C9"$	$x”352” < n < x”433”$
5	$x"7" * n + x"250"$	$x”433” < n < x”514”$
6	$x"A" * n + x"1C7"$	$x”514” < n < x”60E”$
7	$x"f" * n + x"138"$	$x”60E” < n < x”708”$
8	$x"14" * n + x"AD"$	$x”708” < n < x”834”$
9	$x"1A" * n + x"33"$	$x”834” < n < x”8FC”$
10	$x"1F" * n + x"2"$	$x”8FC” < n < x”BB8”$
11	400	$x”BB8” < n$

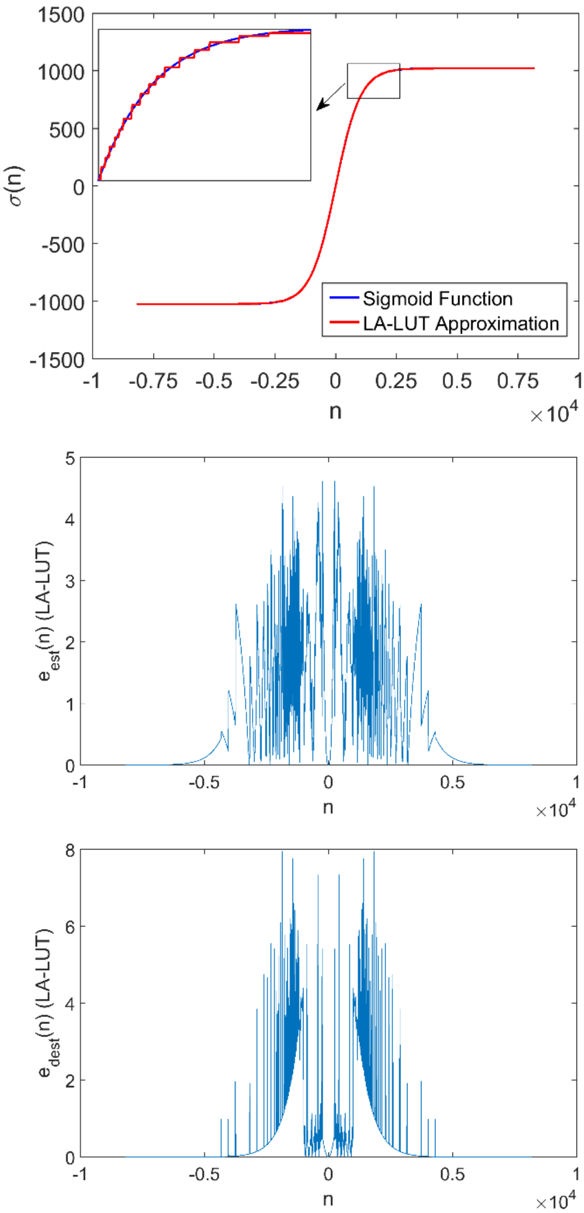

FIGURE 2. LA-LUT feature demonstration. (Top) LA-LUT approximation in comparison to the main sigmoid function, (Middle) Estimation error e_{est} (n), (Bottom) Derivative error e_{dest} (n).

A. FEEDFORWARD CALCULATIONS

The overall scheme of the proposed architecture is demonstrated in Fig. 4. The MLP NN presented in [41] has three input nodes that receive the clock offset samples $\boldsymbol{d} = [d_n \; d_{n+1} \; d_{n+2}]$. They are passed through a MinMax unit which scales and offsets them. This unit operation can be expressed as:

$$\boldsymbol{y_1} = \boldsymbol{d} - \boldsymbol{offset}, \tag{10}$$

$$\boldsymbol{y_2} = \boldsymbol{y_1} \times \boldsymbol{gain}, \tag{11}$$

$$\boldsymbol{X_{p1}} = \boldsymbol{y_2} + \boldsymbol{y_{min}}. \tag{12}$$

In the next step, the first layer weights ($\boldsymbol{IW_1}$) and biases ($\boldsymbol{b_1}$) should be applied to the unit output to form the stimulation of the activation functions:

$$\boldsymbol{X_{p2}} = \boldsymbol{IW_1}^{\boldsymbol{T}} \times \boldsymbol{X_{p1}}, \tag{13}$$

$$\boldsymbol{X_{p3}} = \boldsymbol{b_1} \times \boldsymbol{X_{p2}}. \tag{14}$$

The results are passed through the hardware realization of sigmoid function:

$$\boldsymbol{X_{p4}} = \sigma_{est}\left(\boldsymbol{X_{p3}}\right). \tag{15}$$

Afterward, each neuron output is multiplied in the corresponding weights ($\boldsymbol{IW_2}$) and the second layer bias (b) is added to the result:

$$X_{p5} = \boldsymbol{IW_2}^{\boldsymbol{T}} \times \boldsymbol{X_{p4}}, \tag{16}$$

$$X_{p6} = b + X_{p5}. \tag{17}$$

Eventually, the results are passed through a reverse MinMax unit to achieve the predicted clock offset d_{n+3}:

$$y_3 = X_{p6} - y_{min}, \tag{18}$$

$$y_4 = y_3 \times (\frac{1}{gain}), \tag{19}$$

$$d_{n+3} = y_4 + offset. \tag{20}$$

Note that the network is trained offline. Therefore, the first and second layer weights, biases and coefficients of MinMax units are predetermined and stored in LUTs of the proposed architecture.

B. PROPOSED ARCHITECTURES FOR SIGMOID HARDWRE REALIZATION

As mentioned earlier, one of the main goals of the proposed design is the utilization of fixed-point numbers. Therefore, the clock offset samples are scaled and rounded to maintain the network inputs in the range of [0, 1024). A 10-bit representation is adequate for the inputs; however, 16 bits have been employed to cover the probable overheads in the arithmetic procedures. Furthermore, if an arithmetic operation produces overhead bits, the result is truncated by eliminating the least significant bits to stay in the 16-bit frame. Obviously, the truncation causes precision degradation, which is discussed in the next section.

Fig. 5 demonstrates the architecture of the proposed LA-LUT approximation based on the samples of (4). Similar to LUT implementations, 16-bit comparators have been exploited to specify the range of input. The first comparator ($X_{p3i} < x''00FA$) controls the multiplexer and determines the output is whether obtained by the LUT or linear approximation.

Generally, a line can be defined by its slope and y-intercept. These parameters are extracted from Table 1 and stored in a LUT for implementing PLA. A shown in Fig. 6, the 16-bit comparators are also employed in PLA architecture to specify the corresponding line approximation and control the 16-bit multiplexers. In order to reduce the computational overhead, only a multiplier and an adder have been employed. The multiplier receives the stimulation of the sigmoid function, and the multiplexer determines the slope of the approximation line. The multiplication result is fed to the adder to be aggregated with the corresponding y-intercept to produce the approximation output.

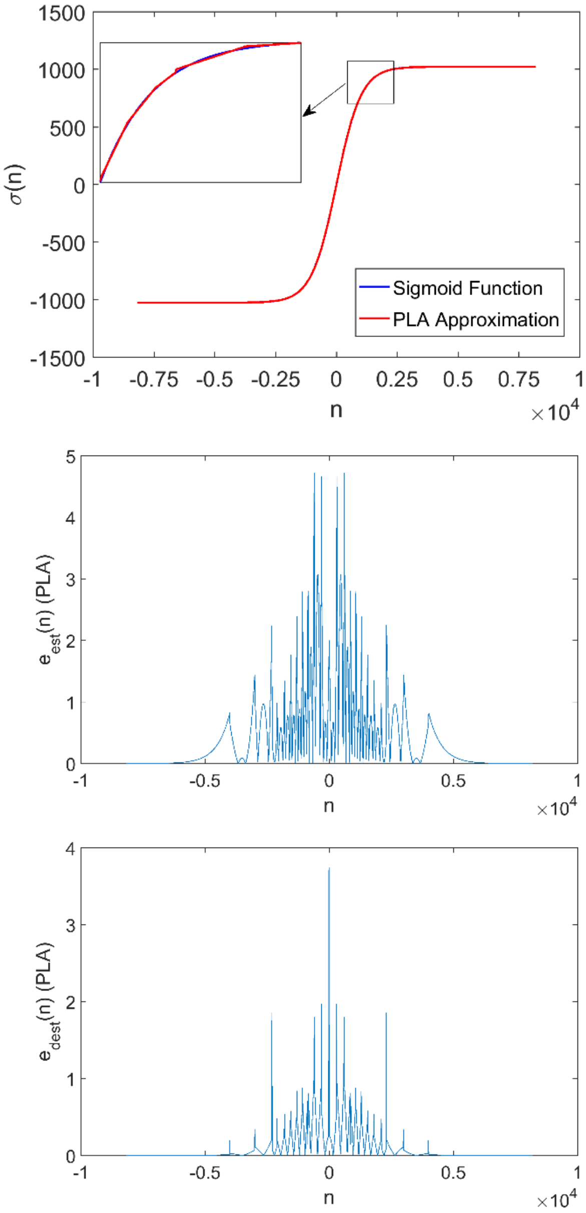

FIGURE 3. PLA feature demonstration. (Top) PLA approximation in comparison to the main sigmoid function, (Middle) Estimation error e_{est} (n), (Bottom) Derivative error e_{dest} (n).

It should be noted that although the architecture of LA-LUT seems more straightforward than the PLA's, the high number of its comparators can consume a considerable amount of hardware resources. In section IV, the overall resource usages of these two architectures are compared to each other to determine the most suitable design.

C. DEFENSE ALGORITHM MODIFICATION

The estimated clock offset ($\boldsymbol{d_{n+3}}$) has to be passed through a TSA detection algorithm to mitigate the probable effects of the attack. The algorithm proposed in [41] is based on high precision estimations of the software version of MLP NN. Achieving the required precision of [41] consumes

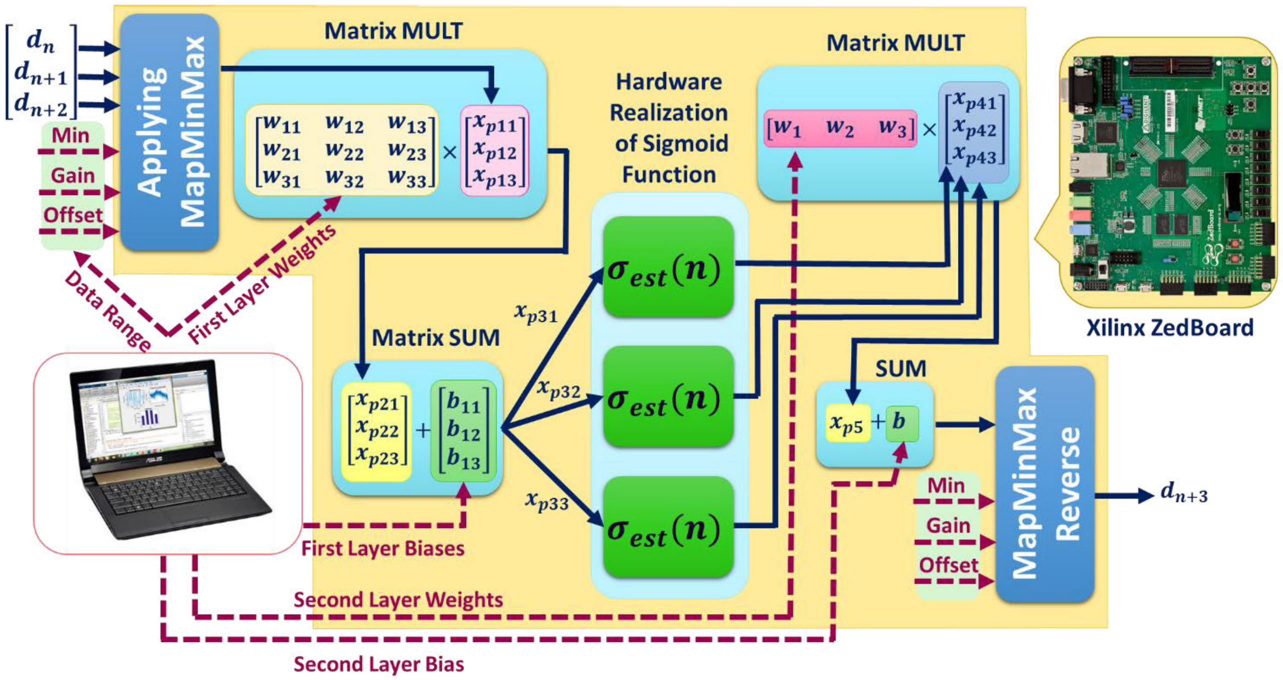

FIGURE 4. Proposed architecture for a N(3,3,1) MLP NN.

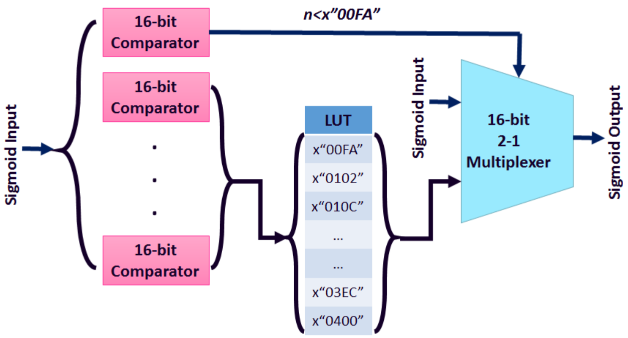

FIGURE 5. Architecture of the proposed LA-LUT.

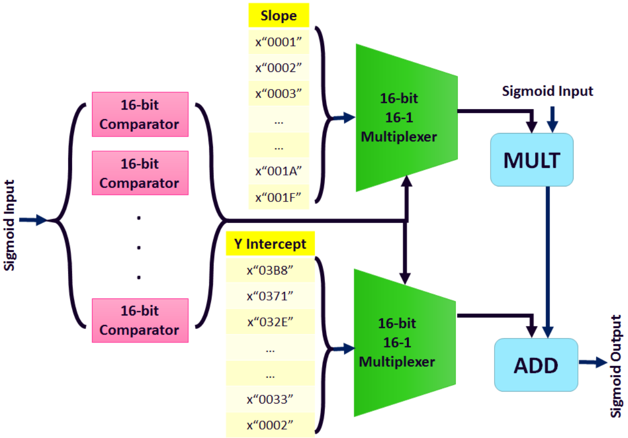

FIGURE 6. Architecture of the proposed PLA.

a considerable amount of hardware resources, which cannot be implemented on the same chip as SDR. Therefore, trade-offs have to be made to reduce the computational resource usage while maintaining a suitable precision, such as sigmoid approximation, rounding, and truncation. The added inaccuracies affect the error-sensitive defense algorithm of [41]. Consequently, the algorithm is modified as presented in Fig. 7 to tolerate reasonable amounts of inaccuracies.

According to [5], TSA has two main behaviors. The first type of TSA is identified by an abrupt change in clock offset information, while the second type includes a gradual modification of the clock offset information. The algorithm of [41] mainly relies on the estimation errors to detect and mitigate the attacks. In the modified version, the differences between the estimated and clock offset samples are also exploited to reduce the sensitivity. Furthermore, another correction coefficient called correction sum is added to mimic the behavior of added signal in the second type of TSA attack. Moreover, the attack declaration threshold can be determined by the application. Section IV demonstrates the evaluation results of the modified algorithm in normal and under attack conditions.

III. ARCHITECTURE ASSESSMENT

This section is dedicated to the performance evaluation and resource usage of the proposed architectures. Furthermore, the provided protection of each design is assessed through two typical TSAs. Both proposed architectures are implemented on Xilinx XC7Z020, which is the main chip of Xilinx ZedBoard. This chip is a popular choice for NN and SDR implementations; thus, many SDRs can employ the proposed designs based on their specifications and required precision.

```
ALGORITHM: TSA Detection and Mitigation (Modified)
Initialize
$corr_{coeff} = 0$;
$corr_{sum} = 0$;
for (each navigation solution extraction)
Calculate NN estimation error as:
$e_{NN}(n+3) = d(n+3) - d_{tu}(n+3)$,
and NN estimation and main clock offset differences as:
$diff_{NN}(n+3) = d(n+3) - d_{corr}(n+2)$,
$diffs_{s}(n+3) = d_{tu}(n+3) - d_{tu}(n+2)$,
      if (clock offset is updating)
         store(all correction coefficients);
        $d_{corr}(n+3) = d_{tu}(n+3) + corr_{coeff}(n+2) + corr_{sum}(n+2)$;
      else
        if ($e_{NN}(n+3) > corr_{coeff}(n+2)$)
           $corr_{coeff}(n+3) = e_{NN}(n+3)$;
        elseif ($diffs_{s}(n+3) > diff_{NN}(n+3)$)
           $corr_{sum}(n+3) = corr_{sum}(n+2) + (diffs_{s}(n+3) - diff_{NN}(n+3))$;
        else
          $corr_{coeff}(n+3) = corr_{coeff}(n+2)$
          $corr_{sum}(n+3) = corr_{sum}(n+2)$;
        end
      $d_{corr}(n+3) = d_{tu}(n+3) + corr_{coeff}(n+2) + corr_{sum}(n+2)$;
      if $corr_{coeff}(n+3) >$ attack declaration threshold
         Declare first type TSA;
      elseif $corr_{sum}(n+3) >$ attack declaration threshold
        Declare second type TSA;
     end
    end
end
```

FIGURE 7. Pseudocode of the modified defense algorithm.

The evaluation dataset has been recorded on April 24, 2014, at Valiasr Street, Tehran, Iran. The sampling frequency of this stationary receiver has been equal to 5.7143 MHz, and the dataset duration is 32.5 seconds, which is adequate for the test. The reason behind the adequacy is the employment of a temperature-compensated crystal oscillator (TCXO) in the GPS receiver board. According to [7], TCXO is not an accurate crystal oscillator; thus, in normal conditions, the receiver has to update the clock and its offset in almost 20-second periods. Regarding the clock update routine, a 32.5-second dataset contains all of the required information of clock offset behavior. The update behavior will be discussed more in upcoming subsections.

In the first subsection, computational resource utilization is discussed, while the second and third subsections contain the evaluation results in the normal and under-attack conditions, respectively.

A. COMPUTATIONAL RESOURCE UTILIZATION

The proposed architectures are implemented by ISE Design Suite 14.2 and VHDL language. Three.txt files are employed to feed the clock offset information to the implemented architecture. These files are structured as:

	First design input	Second design input		Last design input
First file	d_1	d_2	$\ldots$	d_{n-2}
Second file	d_2	d_3	$\ldots$	d_{n-1}
Third file	d_3	d_4	$\ldots$	d_n

In which d represents the clock offset information. The results are also stored as a.txt file. Matlab R2016a has been exploited to organize the input files and display the output of the implemented design.

The sigmoid approximations are implemented based on the schemes presented in Figures 5 and 6, and the results of advanced HDL synthesis for a single neuron activation function are depicted in Fig. 8. According to this figure, the number of exploited elements in the LA-LUT architecture is more than PLA. However, the device utilization report determines which design has been consumed more resources.

The overall design implementation is conducted based on Fig. 4. Therefore, three replicas of each activation function are generated to mimic the behavior of the sigmoid function. The advanced HDL synthesis results and device resource utilization are demonstrated in Tables 2 and 3.

Regarding the results of Table 2, the number of arithmetic operators is almost the same; however, PLA has relied more on arithmetic resources than LA-LUT. The main difference between PLA and LA-LUT is the utilization of registers, comparators, and multiplexers. The register utilization of PLA is almost three times higher than LA-LUT, which is a result of employing one adder and multiplier for the implementation of linear approximations. On the other hand, the LA-LUT mainly relies on comparators and multiplexers, which explains their high utilization amount in comparison to PLA. Fig. 8 also confirms this contrast between the two architectures.

Table 3 represents the device resource utilization and total available resources. In terms of arithmetic operations performed by DSP48E1S cores, both designs almost stand in the same place. A similar conclusion can be made about slice LUT utilization, although the slice register usage in PLA design is considerably higher than LA-LUT. Regarding the available resources, both architectures are consumed less than 0.3% of slice registers, 5% of slice LUTs, and 8% of DSP48E1Ss. Therefore, it can be stated that both architectures are lightweight and can be employed in FPGA-based SDR implementations.

The maximum operating frequency of the architecture is 71.143 MHz for LA-LUT and 53.752 MHz for PLA. The navigation data frequency is equal to 50 Hz, which is way lower than the operating frequency of the architecture. Therefore, both architectures can be exploited as real-time solutions.

B. PERFORMANCE OF THE ALGORITHM IN NORMAL CONDITIONS

The NN is trained by a generated dataset that has a similar slope as recorded data, and the training process is conducted

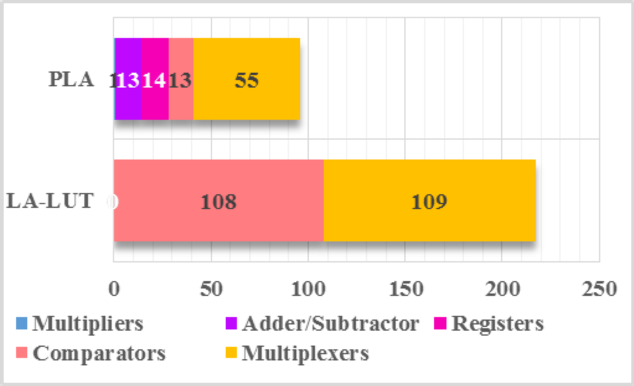

FIGURE 8. Resource usage of sigmoid implementation based on PLA and LA-LUT approximations.

TABLE 2. Overall advanced HDL synthesis results.

Macro Statics	PLA	LA-LUT
Multipliers	**19**	16
Adders/Subtractors	**91**	61
Counters	**9**	0
Registers	**307**	118
Comparators	79	**364**
Multiplexers	242	**407**
XORs	**19**	16

TABLE 3. Device resource utilization.

7Z020clg484	PLA	LA-LUT	Available
Slice Registers	**241**	85	106400
Slice LUTs	2480	**2875**	53200
BUFG/BUFGCTRLS	1	1	32
DSP48E1S	**18**	15	220

offline via Matlab R2016a software. The slope and the behavior of the clock offset trend are highly dependent on the quality of the crystal oscillator. Therefore, it can be stated that as long as the oscillator is not changed, there is no need to retrain.

In the first step of design evaluation, the architecture and the modified algorithm are tested in normal conditions. The assessment results are demonstrated in Figures 9 and 10. The clock offset information of the recorded dataset is displayed on the top panel of both figures. The bottom panel demonstrates the NN estimation error.

The clock offset trend has experienced an abrupt reduction near the 30th sample due to the receiver's clock update. In other words, when the clock offset passes a certain threshold, the receiver's clock is updated regarding the offset, and the clock offset is reduced based on the new clock. This routine is performed periodically to maintain the correct time and hold the clock offset in a predetermined range. Therefore, only one period is selected to evaluate both designs.

According to Fig. 9, the PLA approach has been followed the clock offset trend accurately. However, the LA-LUT approximation has slightly deviated from the authentic trend around the 40th sample, which caused an error near $15\mu s$. Although this deviation is under the attack detection threshold, it certainly affects the prediction accuracy and overall root mean square errors (RMSEs), as demonstrated in the second column of Table 4. In normal conditions, the PLA approach has the advantage of accuracy and precision. In the next subsection, both methods will be evaluated in the presence of two TSAs.

C. EFFICIENCY OF DESIGN IN PRESENCE OF TSA

According to [5], two types of TSA are more likely to happen. The first type is identified by an abrupt modification in the clock offset trend, while the second type of TSA consists of gradual alterations. The attacks are generated by altering the pseudoranges of each in-sight satellite. Therefore, the position of the receiver remains intact, but the clock offset information is changed.

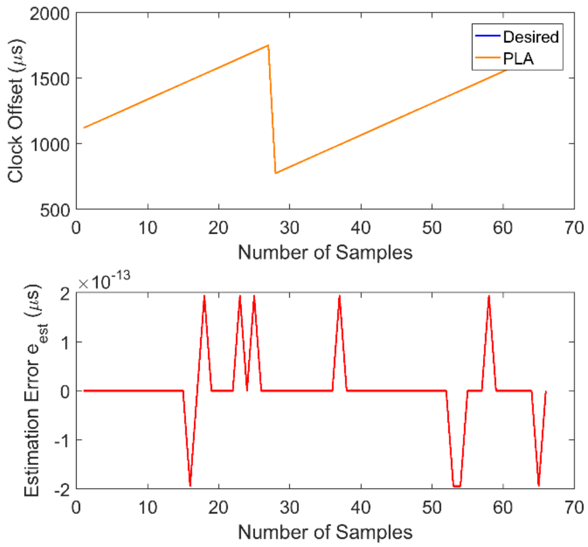

FIGURE 9. Performance of PLA in normal conditions.

The detection and mitigation performance of the PLA and LA-LUT approaches are exhibited in Fig. 11 in the presence of the first type of TSA. An abrupt change in the top panel is observed around the 10th sample, which is an indication of the attack type. The bottom panel demonstrates the NN estimation error e_{NN}. It can be observed that PLA and LA-LUT have mitigated the effects of the attack. According to Figures 2 and 3, the precision of PLA sigmoid approximation is higher than LA-LUT; thus, the PLA approach has higher accuracy compared to the LA-LUT, which can be observed in the bottom panel. It should be noted that although

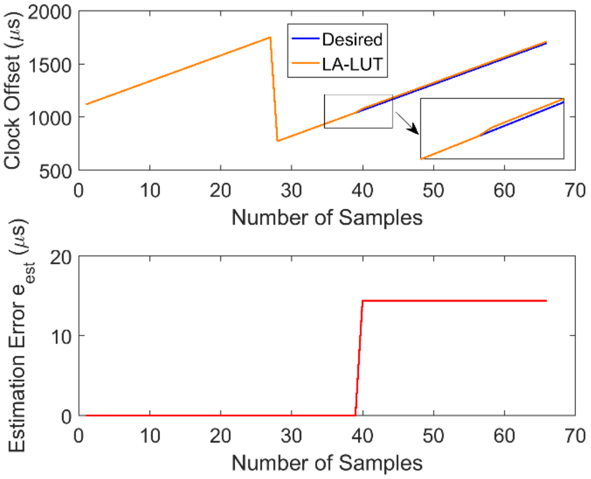

FIGURE 10. Performance of LA-LUT in normal conditions.

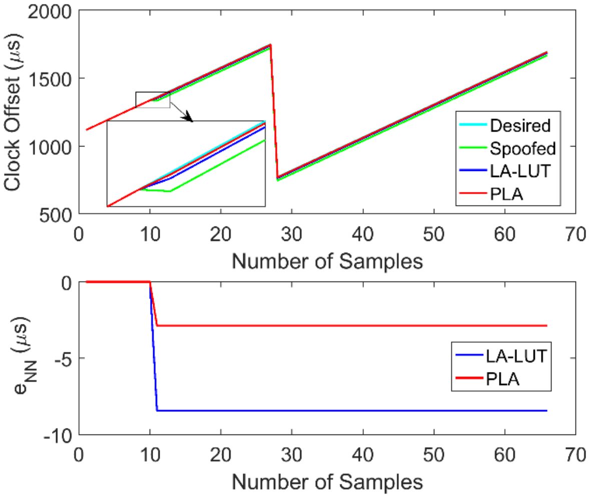

FIGURE 11. PLA and LA-LUT responses in the presence of first type TSA.

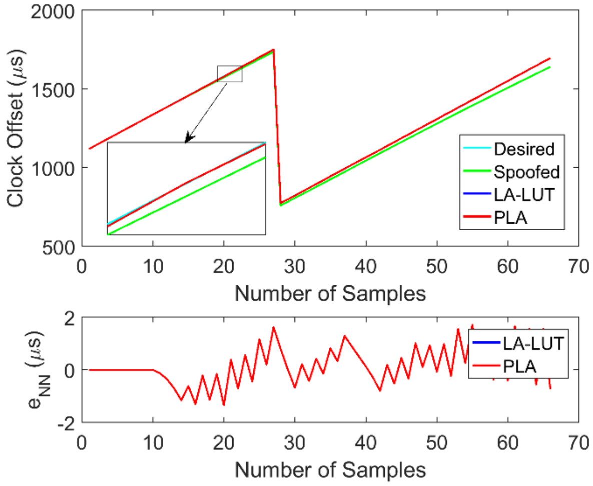

FIGURE 12. PLA and LA-LUT responses in the presence of second type TSA.

TABLE 4. RMSEs of the proposed architectures in comparison to software implementations (expressed in μs).

Detection and mitigation algorithm	Normal conditions	First type TSA	Second type TSA	SW/HW
MLP NN with sigmoid function [41]	7.17×10^{-14}	1.03	0.38	SW
MLP NN with rounded and truncated weights and biases	7.17×10^{-14}	2.66	0.77	SW
MLP NN with PLA (proposed)	7.17×10^{-14}	2.63	0.77	HW
MLP NN with LA-LUT (proposed)	9.20	7.76	0.77	HW

the receiver's clock update is similar to the first type of TSA, the algorithm has stored all of the correction coefficients and prevented the network from misdetection of the situation. The same conclusion can be stated for attack detection in normal conditions.

In the second type of TSA, the modifications are made in the clock offset trend slowly and gradually; therefore, attack detection and mitigation are very challenging. With a degradation in accuracy and precision of MLP NN, the algorithm presented in [41] has not been able to mitigate the effects of the attack. Therefore, the differentiations have been exploited to reduce the sensitivity to approximation errors, as stated in Fig. 7. This factor has significantly reduced the sensitivity to the point that both LA-LUT and PLA have similar performances in the presence of the second type of TSA, as shown in Fig. 12.

The RMSE of each design is compared to the software implemented approach of [41] and an MLP NN with rounded and truncated weights and biases in Table 4. These comparisons are made to demonstrate the effects of roundings, truncations, and sigmoid function approximations. According to the table, roundings and truncations of weights and biases have almost doubled the RMSEs in comparison to [41]. The proposed hardware architectures based on clock offset monitoring are the first ones of their kind; thus, there is no other hardware implementation to compare the results. The RMSEs of the PLA approach are close to the rounded and truncated version. Therefore, it can be concluded that the excess error is mainly caused by rounding and truncation of the weights and biases of the network. On the other hand, although the RMSE of LA-LUT is equal to PLA's in the second type of attack, it has higher RMSEs compared to PLA in two other cases.

IV. CONCLUSION

GPS signals provide position, velocity, and time information or PVT solutions for various users. The position and velocity information is vital for vehicles' navigation; however, the stationary users have a different implication. Considering the fact that the position of stationary receivers is constant and their velocity is zero, the time information has a high significance in this type of receiver. These receivers are generally exploited in critical infrastructures for accurate time measurement; therefore, any interference can result in disastrous incidents. One of these intentional insecurities is the Time synchronization attack which manipulates the clock offset information of the receivers while keeping the receiver's position constant.

Although various software-based countermeasures have been suggested to mitigate the attack consequences, only a few hardware-based protection algorithms are proposed for FPGA-based SDRs. Therefore, this research contributed two different hardware architectures to secure the receivers against TSAs. The first proposed design is based on LUTs, which are very popular for sigmoid approximation, and the second is a high precision PLA. Both architectures are implemented on Xilinx ZedBoard with different resource exploitations. The LA-LUT design has lower precision in comparison to PLA and consumes more logic resources, such as multiplexers and comparators. On the other hand, PLA is mostly arithmetic-based and consumes more slices of DSP48E1S; thus, it is 57% more accurate than LA-LUT. Both designs are lightweight and real-time, and they can be selected for SDRs' protection based on the accuracy or resource utilization priorities.

Concerning the nature of TSA, the proposed designs can mitigate the attack effects on the clock offset information. The knowledge of MLP NN and the modified detection and mitigation algorithm is the basis of this method. However, the network performance should be studied more with new attacks. The new ways of TSA generation will be studied as future works to create maximum protection against TSAs and timing threats. Furthermore, the possibility of applying new structures of NN to mitigate the attack effects will be investigated in the next steps of this research.

REFERENCES

[1] S. Siamak, M. Dehghani, and M. Mohammadi, "Dynamic GPS spoofing attack detection, localization, and measurement correction exploiting PMU and SCADA," *IEEE Syst. J.*, vol. 15, no. 2, pp. 2531–2540, Jun. 2021.

[2] D. Yuan, H. Li, F. Wang, and M. Lu, "A GNSS acquisition method with the capability of spoofing detection and mitigation," *Chin. J. Electron.*, vol. 27, no. 1, pp. 213–222, Jan. 2018.

[3] C. Bonebrake and L. Ross O'Neil, "Attacks on GPS time reliability," *IEEE Secur. Privacy*, vol. 12, no. 3, pp. 82–84, May 2014.

[4] M. R. Mosavi, A. R. Baziar, and M. Moazedi, "De-noising and spoofing extraction from position solution using wavelet transform on stationary single-frequency GPS receiver in immediate detection condition," *J. Appl. Res. Technol.*, vol. 15, no. 4, pp. 402–411, Aug. 2017.

[5] J. Lee, A. F. Taha, N. Gatsis, and D. Akopian, "Tuning-free, low memory robust estimator to mitigate GPS spoofing attacks," *IEEE Control Syst. Lett.*, vol. 4, no. 1, pp. 145–150, Jan. 2020.

[6] E. Schmidt, J. Lee, N. Gatsis, and D. Akopian, "Rejection of smooth GPS time synchronization attacks via sparse techniques," *IEEE Sensors J.*, vol. 21, no. 1, pp. 776–789, Jan. 2021.

[7] P. N. Misra, "The role of the clock in a GPS receiver," *GPS World*, vol. 6, pp. 60–66, Apr. 1996.

[8] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: A Single Frequency Approach*. Boston, MA, USA: Springer, 2007.

[9] X. Jiang, J. Zhang, B. Harding, J. J. Makela, and A. D. Dominguez-Garcia, "Spoofing GPS receiver clock offset of phasor measurement units," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3253–3262, Aug. 2013.

[10] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016.

[11] E. Schmidt, Z. A. Ruble, D. Akopian, and D. J. Pack, "A reduced complexity cross-correlation interference mitigation technique on a real-time software-defined radio GPS L1 receiver," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2018, pp. 931–939.

[12] Q. Lu, X. Feng, and C. Zhou, "A detection and weakening method for GNSS time-synchronization attacks," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19069–19077, Sep. 2021.

[13] S. Bhamidipati, K. J. Kim, H. Sun, and P. V. Orlik, "Artificial-intelligence-based distributed belief propagation and recurrent neural network algorithm for wide-area monitoring systems," *IEEE Netw.*, vol. 34, no. 3, pp. 64–72, May 2020.

[14] Y. Hu, S. Bian, B. Li, and L. Zhou, "A novel array-based spoofing and jamming suppression method for GNSS receiver," *IEEE Sensors J.*, vol. 18, no. 7, pp. 2952–2958, Apr. 2018.

[15] Y. Hu, X. Dong, Z. Wu, and Z. Shi, "Spoofing mitigation for GPS receiver based on array antenna using cross-correlation of received signals of each element," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 7295–7300.

[16] C. Fernández-Prades, J. Arribas, and P. Closas, "Robust GNSS receivers by array signal processing: Theory and implementation," *Proc. IEEE*, vol. 104, no. 6, pp. 1207–1220, Jun. 2016.

[17] M. R. Mosavi, A. Tabatabaei, and M. J. Zandi, "Positioning improvement by combining GPS and GLONASS based on Kalman filter and its application in GPS spoofing situations," *Gyroscopy Navigat.*, vol. 7, no. 4, pp. 318–325, Oct. 2016.

[18] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "GPS spoofing detection via dual-receiver correlation of military signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2250–2267, Oct. 2013.

[19] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1073–1090, Apr. 2013.

[20] K. Ghorbani, N. Orouji, and M. R. Mosavi, "Navigation message authentication based on one-way hash chain to mitigate spoofing attacks for GPS l1," *Wireless Pers. Commun.*, vol. 113, no. 4, pp. 1743–1754, Aug. 2020.

[21] E. Schmidt, Z. Ruble, D. Akopian, and D. J. Pack, "Software-defined radio GNSS instrumentation for spoofing mitigation: A review and a case study," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2768–2784, Aug. 2019.

[22] K. Charqane, A. Defina, F. Dominici, G. Marucco, and P. Mulassano, "AT-SURF and SAT-SURFER: Novel hardware and software platform for research and education on satellite navigation," in *Proc. 22nd Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, May 2009, pp. 1937–1942.

[23] D. F. M. Cristaldi, D. Margaria, and L. Lo Presti, "A multifrequency low-cost architecture for GNSS software receivers," in *Proc. Int. Tech. Meeting Inst. Navigat.*, Jan. 2010, pp. 679–687.

[24] J. S. Silva, P. F. Silva, A. Fernandez, J. Diez, and J. F. M. Lorga, "Factored correlator model: A solution for fast, flexible, and realistic GNSS receiver simulations," in *Proc. 20th Int. Tech. Meeting Satell. Division The Inst. Navigat. (ION GNSS)*, 2007, pp. 2676–2686.

[25] E. Schmidt, D. Akopian, and D. J. Pack, "Development of a real-time software-defined GPS receiver in a labVIEW-based instrumentation environment," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 9, pp. 2082–2096, Sep. 2018.

[26] E. Schmidt and D. Akopian, "Exploiting acceleration features of LabVIEW platform for real-time GNSS software receiver optimization," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS+)*, Nov. 2017, pp. 3694–3709.

[27] C. Fernandez-Prades, J. Arribas, P. Closas, C. Aviles, and L. Esteve, "GNSS-SDR: An open source tool for researchers and developers," in *Proc. 24th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2011, pp. 780–794.

[28] A. Soghoyan, A. Suleiman, and D. Akopian, "A development and testing instrumentation for GPS software defined radio with fast FPGA prototyping support," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 8, pp. 2001–2012, Aug. 2014.

[29] Y. Ng and G. X. Gao, "GNSS multireceiver vector tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 5, pp. 2583–2593, Oct. 2017.

[30] L. L. Presti, P. di Torino, E. Falletti, M. Nicola, and M. T. Gamba, "Software defined radio technology for GNSS receivers," in *Proc. IEEE Metrol. Aerosp. (MetroAeroSpace)*, May 2014, pp. 314–319.

[31] P.-L. Normark and C. Ståhlberg, "Hybrid GPS/Galileo real time software receiver," in *Proc. 18th Int. Tech. Meeting Satell. Division The Inst. Navigat. (ION GNSS)*, Sep. 2005, pp. 1906–1913.

[32] J. Dampf, T. Pany, W. Bär, J. Winkel, C. Stöber, K. Fürlinger, P. Closas, and J. A. Garcia-Molina, "More than we ever dreamed possible: Processor technology for GNSS software receivers in the year 2015," *Inside GNSS*, vol. 10, no. 4, pp. 62–72, 2015.

[33] G. Heinrichs, M. Restle, C. Dreischer, and T. Pany, "NavX–NSR—A novel Galileo/GPS navigation software receiver," in *Proc. 20th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2007, pp. 1329–1334.

[34] C. Stöber, M. Anghileri, A. S. Ayaz, D. Dötterböck, I. Krämer, and V. Kropp, "ipexSR: A real-time multi-frequency software GNSS receiver," in *Proc. ELMAR*, 2010, pp. 407–416.

[35] J. Tian, W. Ye, S. Lin, and Z. Hua, "SDR GNSS receiver design over standalone generic TI DSP platform," in *Proc. IEEE 10th Int. Symp. Spread Spectr. Techn. Appl.*, Aug. 2008, pp. 42–47.

[36] E. G. Lightsey, T. E. Humphreys, J. A. Bhatti, A. J. Joplin, B. W. O'Hanlon, and S. P. Powell, "Demonstration of a space capable miniature dual frequency GNSS receiver," *Navigat. J. Inst. Navigat.*, vol. 61, no. 1, pp. 53–64, 2014.

[37] R. Calvo-Palomino, A. Bhattacharya, G. Bovet, and D. Giustiniano, "Short: LSTM-based GNSS spoofing detection using low-cost spectrum sensors," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Aug. 2020, pp. 273–276.

[38] A. Fridman and S. Semenov, "System-on-chip FPGA-based GNSS receiver," in *Proc. East-West Design Test Symp. (EWDTS)*, Sep. 2013, pp. 1–7.

[39] M. S. Meraz, J. M. C. Arvizu, and A. J. A. Cruz, "GNSS receiver based on a SDR architecture using FPGA devices," in *Proc. IEEE Electron., Robot. Automot. Mech. Conf.*, Nov. 2011, pp. 383–388.

[40] G. Artaud, L. Ries, M. Monnerat, H. Al-Bitar, F. Legrand, and M. Weyer, "Development of a flexible real time GNSS software receiver," in *Proc. 5th ESA Workshop Satell. Navigat. Technol. Eur. Workshop GNSS Signals Signal Process. (NAVITEC)*, Dec. 2010, pp. 1–8.

[41] N. Orouji and M. R. Mosavi, "A multi-layer perceptron neural network to mitigate the interference of time synchronization attacks in stationary GPS receivers," *GPS Solutions*, vol. 25, no. 3, p. 84, Jul. 2021.

[42] P. Axelrad and R. G. Brown, "GPS navigation algorithms," in *Global Positioning System: Theory and Applications*, vol. 1, Washington, DC, USA: American Institute of Aeronautics and Astronautics, 1996, pp. 409–433.

[43] D. G. Luenberger, "Observers for multivariable systems," *IEEE Trans. Autom. Control*, vol. AC-11, no. 2, pp. 190–197, Apr. 1966.

[44] V. Tiwari and N. Khare, "Hardware implementation of neural network with sigmoidal activation functions using CORDIC," *Microprocess. Microsyst.*, vol. 39, no. 6, pp. 373–381, 2015.

[45] I. Campo, R. Finker, J. Echanobe, and K. Basterretxea, "Controlled accuracy approximation of sigmoid function for efficient FPGA-based implementation of artificial neurons," *Electron. Lett.*, vol. 49, no. 25, pp. 1598–1600, Dec. 2013.

[46] Z. Hajduk, "High accuracy FPGA activation function implementation for neural networks," *Neurocomputing*, vol. 247, pp. 59–61, Jul. 2017.

[47] Z. Hajduk, "Reconfigurable FPGA implementation of neural networks," *Neurocomputing*, vol. 308, pp. 227–234, Sep. 2018.

[48] I. Tsmots, O. Skorokhoda, and V. Rabyk, "Hardware implementation of sigmoid activation functions using FPGA," in *Proc. IEEE 15th Int. Conf. Exper. Designing Appl. CAD Syst. (CADSM)*, Feb. 2019, pp. 34–38.

[49] V. Beiu, J. Peperstraete, J. Vandewalle, and R. Lauwereins, "Close approximations of sigmoid functions by sum of step for VLSI implementation of neural networks," *Sci. Ann. Informat.*, vol. 40, no. 1, pp. 1–20, 1994.

NILOOFAR OROUJI received the B.S. degree in electronic engineering from the K.N. Toosi University of Technology, Tehran, Iran, in 2014, and the M.S. degree in digital electronic systems from the Iran University of Science and Technology, Tehran, in 2017, where she is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. Currently, she is with the Universidad Politécnica de Madrid (UPM). Her research interests include specialized architecture design, system security, and novel co-processors.

MOHAMMAD REZA MOSAVI received the B.S., M.S., and Ph.D. degrees in electronic engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 1997, 1998, and 2004, respectively. He is currently a Faculty Member (Full Professor) with the Department of Electrical Engineering, IUST. He is the author of more than 450 scientific publications in journals and international conferences and 12 academic books. His research interests include circuits and systems design. He is also the Editor-in-Chief of *Iranian Journal of Marine Technology* and an Editorial Board Member of *Iranian Journal of Electrical and Electronic Engineering*.

DIEGO MARTÍN received the B.Sc. degree in computer engineering and the M.Sc. degree in computer science from the Department of Informatics, University Carlos III of Madrid, Spain, and the Ph.D. degree from the University Carlos III of Madrid, in 2012. He is currently a Lecturer with the Department of Telematics, Technical University of Madrid (UPM). His main research subjects, within the GISAI groups at UPM, are the Internet of Things, cyber physical systems, physically unclonable functions, blockchain, knowledge management, information retrieval, and research methods.