International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Twin Error Detection in Luhn's Algorithm

Waweru Kamaku[1], Wangeci Wachira[2]

Department of Pure and Applied Mathematics, Jomo Kenyatta University of Agriculture and Technology.

Abstract: ***The Luhn formula or algorithm was created by Han's Peter Luhn in the 1950's to solve storage and retrieval problems. This algorithm has been given world wide application such as in the banking industry, the mobile phone industry among others. For instance, credit cards numbers that uniquely identifies an account to its respective owner uses the Luhn formula. Insecurity and fraud cases associated with credit cards has necessitated the need to analyze the algorithm's error detection and correction capabilities. It is known that Luhn formula does not detect twin errors*** $22 \rightarrow 55$, $33 \rightarrow 66$ ***and*** $44 \rightarrow 77$. ***This paper analyzes the detection of these twin errors and discusses the key considerations in algorithm development to avoid such an error.***

Keywords: Algorithm, Error detection, Error Correction, Twin error, Luhn Algorithm

1. Introduction

An algorithm is a finite sequence of precise instructions followed when performing a computation or solving a problem. All algorithms generally share the same properties, that is, they have an input and a corresponding output, must be correct, definite, finite, effective and should be applicable to all problems of the same nature(Rosen, 2012). An error is a deviation from accuracy or correctness usually caused by noisy communication channel such as humans, thermal noise, imperfection in equipments among others. If a code word u is transmitted through a noisy communication channel there is a possibility that a different code word v will be received instead of the original code word u implies that an error(s) exist. Error detection is the identification of errors in a code word of which it may be discarded and request for retransmission made. Error correction is the detection of errors in a code word and reconstruction of the original error free data. According to Gallian (1991), there are several types of errors that occur during data entry such as single errors, transposition errors (adjacent and jump), twin errors, phonetic errors among others. Twin errors are made when a pair of similar digits are replaced with another pair, that is aa is replaced with a pair of digits such as bb. For example $22 \rightarrow 55$(Eric, 2017). If a and b are integers and m is a positive integer, than a is cogruent to b modulo m if m divides $a-b$ or $m|(a-b)$ or $a-b=mk$ where $k \in \mathbb{Z}$. The notation $a \equiv b \ (mod\ m)$ is uused to indicate that a is congruent to b modulo m where m is referred as the modulus while $a \equiv b \ (mod\ m)$ is referred to as the congruence (Raymond, 1986). A prime number p is a number divisible by ± 1 or $\pm p$ while a composite number p is a number that can be written as $p = a.b$ where $1 < a, b < p$ and $a, b \in \mathbb{Z}$. If p is a prime number, integers modulo p consist of the integers $\{0,1,\dots,p-1\}$ with addition and multiplication performed modulo p forms a finite fieldof order p while if p is composite it does not form a finite field since not all integers in the set $\{0,1,\dots,p-1\}$ have inverses. Zero divisors occur if $a, b \in \{0,1,\dots,p-1\}$ and $0 < a, b < p$ then $a \cdot b \equiv 0 \ (mod\ p)$. Checksum is a small sized datum computed form an arbitrary block of digital data while a checkdigit is an alphabet or numeric usually at the end of the code word used for error detection.

2. Validation and Check Digit Calculation Using Luhn Formula

In validation, the formula verrifeis the numbers against its included checkdigit. Counting from the check digit, the rightmost digit, moving left double the value of every even positioned digit. If all the products p are less than 10, calculate $\left(\sum_{i=0}^{k} 2n_{2i} + \sum_{i=0}^{k} n_{2i+1}\right) mod\ 10$, where $n \in \mathbb{Z}_{10}$ and $i, k \in \mathbb{Z}$ but if $\exists\ p \geq 10$ add the digits of the products together to obtain a single digit. That is, if $p = ab$ calculate $a + b = c$ before calculation of the checksum $mod\ 10$.
If $\left(\sum_{i=0}^{k} 2n_{2i} + \sum_{i=0}^{k} n_{2i+1}\right) mod\ 10 = 0$, then codeword is valid, otherwise it is invalid
(Ghadhi, 2017).

If $a_1 a_2 \dots a_n$ is an account number that will have a checkdigit added to it to be of the form $a_1 a_2 \dots a_n x$. Multiply through by 2 every even positioned digits from the rightmost to obtain $p_1 p_2 \dots p_n x$. If $\exists p_i \geq 10$, sum the digits of the product to obtain a single digit then calculate the checkdigit $x = 10 - (\sum_{i=1}^{n} p_i)\ mod\ 10$but if all $p_i < 10$, calculate the checkdigit x directly as $x = 10 - (\sum_{i=1}^{n} p_i)\ mod\ 10$. The check digit will only be 0 if $(\sum_{i=1}^{n} p_i)\ mod\ 10 = 0$ otherwise the checkdigit is in the set $\{1,2,\dots,9\}$.

Ghadhi (2017) analyzed the various types of checkdigit algorithms and observed that the Luhn algorithm will detect 7 out of 10 twin errors excluding $22 \rightarrow 55, 33 \rightarrow 66$ and $44 \rightarrow 77$. That is, replacing any of the pairs by the corresponding twin will produce the same checksum and ultimately the same checkdigit. Thus the error will not be detected.Gallian (1991) gave conditions necessary for detection of different types of errors, for instance, for detection of twin errors $aa \rightarrow bb$ at positions i and $i+1$ the $\gcd(w_i + w_{i+1}, k) = 1$ where k is the modulus. Various checkdigit algorithms have been analyzed over the years but none of those resesearchers heve shown why the Luhn formula fails in detection of some twin errors. What are the weaknesses in the algorithm that makes these 3 twin errors to pass undetected? This paper analyses the Luhn Formula in regards to twin error detection and points out the weaknesses

in the formula that makes those 3 twin errors to pass undetected.

3. Results and Discussion

The following Theorem and corollary states the condition necessary for a twin error to be detected by the Luhn formula.

Theorem 1: Let $a, b \in \mathbb{Z}_{10}, 1 \leq a \leq 4$ and $5 \leq b \leq 9$ and w_i be the weight used. Luhn formula detects all twin errors of the form $aa \rightarrow bb$ if

$$\begin{cases} aw_i - bw_i \neq -3 \text{ when } w_i = 1 \\ aw_i - bw_i \neq 3 \text{ when } w_i = 2 \end{cases}$$

Proof. By contraposition, suppose Luhn formula does not detect all twin errors of the form $aa \rightarrow bb$. This implies that both twins have the same checksum contribution, that is,

$aw_i + aw_j = bw_i + bw_j$

$aw_i - bw_i = bw_j - aw_j$

The table 1.1 is a table of $aw_i - bw_i$ when $w_i = 1$

Table 1.1: Twin Error Detection when $w_i = 1$

$aw_i - bw_i$		5	6	7	8	9
aw_i	1	-4	-5	-6	-7	-8
	2	-3	-4	-5	-6	-7
	3	-2	-3	-4	-5	-6
	4	-1	-2	-3	-4	-5

→

$aw_i - bw_i$	5	6	7	8	9
1	√	√	√	√	√
2	×	√	√	√	√
3	√	×	√	√	√
4	√	√	×	√	√

√ - *Detectable* × - Undetectable

From Table 1.1 above, it is seen that the Luhn formula does not detect twin errors if $aw_i - bw_i = -3$. That is when $a = 2$ and $b = 5$ or when $a = 3$ and $b = 6$ or when $a = 4$and $b = 7$. These are the only twin error undetectable cases in Luhn Algorithm. Hence the contraposition is true that Luhn formula detects twin errors of the form $aa \rightarrow bb$ if $aw_i - bw_i \neq -3$when $w_i = 1$. Without loss of generality, Luhn formula detects twin error of the form $aa \rightarrow bb$ if $bw_j - aw_j \neq -3$where $w_i \neq w_j = 2$. Similarly, if $w_i = 2$, the following is a table of $aw_i - bw_i$. Recall that, $\forall bw_i \geq 10$ compute $bw_i - 9$ if $9 | bw_i$ or compute $bw_i \ mod\ 9$ if $9 \nmid bw_i$ to obtain a single digit.

Table 1.2: Twin Error Detection when $w_i = 2$

$aw_i - bw_i$		1	3	5	7	9
aw_i	2	1	-1	-3	-5	-7
	4	3	1	-1	-3	-5
	6	5	3	1	-1	-3
	8	7	5	3	1	-1

→

$aw_i - bw_i$	1	3	5	7	9
2	√	√	√	√	√
4	×	√	√	√	√
6	√	×	√	√	√
8	√	√	×	√	√

√ - Detectable × - Undetectable

From Table 1.2, it is shown that Luhn formula does not detect twin errors if $aw_i - bw_i = 3$ when $w_i = 2$. That is when $a = 2$ and $b = 5$ or when $a = 3$ and $b = 6$ or $a = 4$and $b = 7$. Thus the contraposition is true that Luhn formula detect twin errors of the form $aa \rightarrow bb$ if $bw_j - aw_j \neq 3$ where $w_j = 1 \neq w_i$.

Corollary 1: Let $a, b \in \mathbb{Z}_{10}$, $5 \leq a \leq 9$ and $1 \leq b \leq 4$ and w_i be the weight used. Luhn formula detects all twin errors of the form $aa \rightarrow bb$ if

$$\begin{cases} aw_i - bw_i \neq 3 \text{ when } w_i = 1 \\ aw_i - bw_i \neq -3 \text{ when } w_i = 2 \end{cases}.$$

The following Proposition disagrees with Gallian (1991) claim that an algorithm will detect twin error if the gcd of the sum of the weights and the modulus is equal to one. Also, the effects of zero divisor on error detection is seen.

Proposition 1: Let w_i and w_{i+1} be weights, $(w_i + w_{i+1}, 10) = 1$ does not guarantee all twin error detection in Luhn algorithm.

Proof. By counter example, let 22 be two adjacent digits in a code which are replaced by 55. In Luhn algorithm $(w_i + w_{i+1}, 10) = 1$since $\gcd(3, 10) = 1$. Let $w_i = 2$ and $w_{i+1} = 1$, but

*Proof.*The proof to be followed from Theorem 1.

$$(2w_i + 2w_{i+1}) - (5w_i + 5w_{i+1}) = 2 \cdot 2 + 2 \cdot 1 - (5 \cdot 2 + 5 \cdot 1) = 4 + 2 - (10 + 5) = 6 - 1 - 5 \equiv 0 \ (mod\ 10)$$

since for any product $p \geq 10$, compute $p - 9$ if $9 | p$ or compute $p \ (mod\ 9)$ if $9 \nmid p$ to obtain a single digit. This applies for digits $33 \rightarrow 66$ and digits $44 \rightarrow 77$. Thus Luhn formula does not guarantee detection of all twin errors even though $(w_i + w_{i+1}, 10) = 1$.

The following Theorem proposes the condition that should be adhered to when choosing digits for a code word in order for it to be twin error free while using the Luhn formula

Theorem 2: Let $a, b \in \mathbb{Z}_{10}$ such that $1 \leq a \leq 4$ and $5 \leq b \leq 9$ and w_i and w_j be the weights. If $(a-b)\cdot(w_i + w_{i+1}) = -(k^2)$, $k \in \mathbb{Z}$ then Luhn formula does not detect twin error of the form $aa \longrightarrow bb$.

Proof.
There are only three instances when $(a-b)(w_i + w_{i+1}) = -(k^2)$. That is when $a = 2, b = 5$ or $a = 3, b = 6$ or when $a = 4, b = 7$. In all these three cases it is shown in Theorem 1 that Luhn formula does not detect twin error of the form $aa \rightarrow bb$.

Theorem 3: Let $a, b \in \mathbb{Z}_{10}$ and w_i and w_{i+1} be the weights where $w_i \neq w_{i+1}$. In Luhn formula $(aw_i + aw_{i+1}) - (bw_i + bw_{i+1}) \neq (a-b)(w_i + w_{i+1})$.

Proof.
By counter example, let $a = 3$ and $b = 6$.
$(aw_i + aw_{i+1}) - (bw_i + bw_{i+1}) \equiv 0\ (mod\ 10)$ since for every $aw, bw \geq 10$, subtract 9 to obtain a single digit. $[(a-b)(w_i + w_{i+1}) \neq 0]$. Thus, $(aw_i + aw_{i+1}) - (bw_i + bw_{i+1})$ is not always equal to $(a-b)(w_i + w_{i+1})$ in Luhn algorithm. That is, Luhn algorithm is not always distributive.

4. Conclusion

The summation of the weights and the modulus being coprime does not grantee that an algorithm will detect all twin errors which disagrees with Gallian (1991) papers. Inorder for a twin error to be detected by the Luhn form one should ensure that the digits and weights chosen should comply with the condition stated. This will reduce chances for a twin error to pass the Luhn formula undetected. Otherwise, there is need for an improved algorithm that captures all the twin errors without restriction on any digit chosen.

5. Recommendation

More analysis can be done on the Luhn Algorithm with respect to the detection of other types of errors.

References

[1] Eric, W. (2017). *Error Correcting Code.* Retrieved from Wolfram: HTTP://www.mathworld.wolfram.com/Error-Correcting Code.HTML. 19/6/2017
[2] Gallian, J. (1991). The Mathematics of Identification Numbers. *The College of Mathematics Journal*, 22(3), 194-202.
[3] Ghadhi, K. (2017). *Check Digits*. HTTP://cosmos.ucdavis.edu/archives/2010/cluster6/ghadhikira.HTML. 19/6/2017
[4] Raymond, H. (1986). *A first Course in Coding Theory.* Oxford: Clarendon Press.
[5] Rosen, K. (2012). *Discrete Mathematics and its Application.* New York: Oxford University Press.