Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 1098 – 1110

doi:10.3906/elk-1306-96

RFID card security for public transportation applications based on a novel neural network analysis of cardholder behavior characteristics

Gürsel DÜZENLİ*
Department of Electronic Engineering, Faculty of Engineering, Sakarya University, Sakarya, Turkey

Received: 11.06.2013 • **Accepted/Published Online:** 16.07.2013 • **Printed:** 10.06.2015

Abstract: This paper proposes a novel approach that applies neural network forecasting to security for closed-loop prepaid cards based on low-cost technologies such as RFID and 1-Wire. The security vulnerability of low-cost RFID closed-loop prepaid card systems originates mostly from the card itself. Criminal organizations counterfeit or clone card data. Although high-security prepaid cards exist, they are often too expensive for transport ticketing, and even their security is not guaranteed for a well-defined period of time. Therefore, data encryption systems are used widely against counterfeiting with success. However, it has not been possible to develop countermeasures with comparable success against cloning. Our proposed security application uses neural network forecasting to determine the recharge day behavioral characteristics of the cardholder and predict the next time the cardholder will recharge their card. Based on the prediction for the recharge time, the expiration date of the low-cost RFID prepaid card is defined, which is a good countermeasure against cloning. FTDNN, LRNN, and NARX network architectures with one hidden layer are considered in this research. The effects of the network architecture, the number of neurons, the training algorithm, and the prediction performance function on the recharge day forecast are investigated. Experimental results confirm the accuracy of the recharge time forecast and confirm countermeasures against cloning. Our proposed security approach with neural network forecasting is applied with success to the Turkish public transport without an online backend system.

Key words: Low-cost, RFID, prepaid card, MIFARE, DST, iButton, security, neural network, NARX

1. Introduction

For the past several years, the use of prepaid cards with radio frequency identification (RFID) technology has grown rapidly as an alternative to credit and debit cards. The exponential growth of RFID prepaid cards brings not only prosperity but also vulnerability, especially for low-cost cards. These low-cost tags have little or no processing and storage capacity. Therefore, it is not possible to implement well-known cryptographic protocols such as 3DES (Triple-Data Encryption Standard) or AES (Advanced Encryption Standard) to increase the tag security. However, RFID cards with simple logical functions as cryptographic protocols are preferred for applications where many cards are needed, such as prepaid telephone cards and transit fare cards, because these cards are inexpensive. However, numerous attacks have challenged security mechanisms for low-cost RFID cards, such as Philips MIFARE RFID tags [1–9], Texas Instruments DST RFID tags [10–12], and Maxim iButtons [13]. Furthermore, the unique serial number (UID) of a chip is initialized only by the manufacturer, but a China-based company produced unlicensed clones of MIFARE chips [7,8]. Highly secure tags have greater processing and storage capabilities. Therefore, these cards are expensive, but they have the capability to be

*Correspondence: gursel@duzenli.net

high-security tags. Certainly, a card's security is not guaranteed for a well-defined period of time. Kasper et al. [14] showed how to fully recover the 3DES key of a high security DESFire tag with a factory-programmed UID from MIFARE using side-channel analysis. Furthermore, Kasper et al. [15] could clone and restore the credit balance of a card to its original state to provide an infinite amount of payments with the high security DESFire tag with 3DES and a DESFire EV1 tag with AES from MIFARE. Although the security level of credit cards is extremely high because they use online backend systems to finish transactions, Murdoch et al. [16] found vulnerabilities in the security standard used to make payments without knowing the card's PIN and to remain undetected while using an online connection to the banking network. These security vulnerabilities show that security with cryptographic protocols is never sufficient to provide high security for tags, especially for a defined time period. The main sources of these security vulnerabilities are the unchangeable cryptographic protocols and authentications. Although it is important to secure a system the first time it is used, it is more important to have the ability to adapt to a higher security level if needed. However, this adaptability is not applicable to public systems where tags and readers must be reprogrammable, which requires a higher initial cost. Our proposed security approach fulfills this need with a completely different form of security, and it is suitable for applications with low-cost technologies, such as RFID and 1-Wire. Our proposed security approach depends on the time of recharge and the cardholder's behavioral characteristics, which are specific for each cardholder and function as a dynamic 'unique identifier' [17]. From these behavioral characteristics, we use neural network (NN) forecasting to predict the card expiration date as a countermeasure against cloning.

NN has been applied successfully as a powerful modeling technique in a wide range of research areas. NN refers to mathematical models based upon the functionality of the human brain. These models contain at least three different layers, input, hidden, and output, and each layer is composed of a number of neurons. NN models are increasingly applied to systems that are highly correlated and frequently assumed to be nonlinear, having unclear relationships and being too complex for other approaches [18–22]. We apply NN forecasting to define a security feature that is unique and dynamic for every cardholder. This approach meets the most demanding requirements for security over a long period of time.

This paper is organized as follows. Section 2 gives a brief overview and a comparison of prepaid cards and their security levels. Our proposed approach to security is described in Section 3. Section 4 focuses on the NN forecasting architecture, while Section 5 discusses the data analysis and the training of the NN forecasting networks. Section 6 presents the simulation results and the discussion of these results, and Section 7 addresses the conclusions.

2. Prepaid cards

Prepaid cards can be divided into two main groups. The first group includes the open-loop prepaid cards (OLPCs). These prepaid cards are used as standard debit cards, but without the need for a bank account. They can be utilized at any retailer that accepts credit/debit cards, and they can be used for receiving direct deposits or making withdrawals at ATM machines. The second type, the closed-loop prepaid cards (CLPCs), are limited for use with specific merchants or purchases, such as prepaid telephone cards and transit fare cards.

An important difference between the OLPCs and the CLPCs is the security mechanism of the card. Because the OLPCs are used in banking systems, their security is standardized and identical worldwide. These security standards and other supporting guidance documents are generated by EMV (Europay, MasterCard, and VISA) and the Payment Card Industry Security Standards Council (PCI SSC) to provide the greatest level of security for the whole system. EMV and PCI SSC seek to define a common set of high-level security standards

for the OLPCs and the system. However, the CLPC cards are used by specific merchants and purchasers, and there is no widespread public examination, such as that of EMV or PCI SSC, to provide high-level security for the CLPCs and their applications. Only the security developed by the card manufacturer protects the card data. However, the manufacturers conceal their cryptographic algorithm and design to provide security. This is known as 'security by obscurity', but the details of this system will eventually become public information, which was the case for low-cost MIFARE RFID tags, DST RFID tags, and iButtons. Nevertheless, there are a great number of applications for these cards, each with its own security, and new applications can arise at any time because these cards are inexpensive.

Another important difference between the OLPCs and the CLPCs is the connection from the terminals to the backend system. The OLPCs are online systems, and their terminals must be connected online to their backend systems to complete the transactions. Therefore, the online systems are using fraud detection by monitoring card transactions and analyzing the data to detect unusual behavior. However, most CLPCs are offline systems, and their terminals are offline from their backend systems. Transactions are stored in the terminal unit they are transferred to the backend system. Therefore, the CLPCs are not as secure as the OLPCs. Our proposed neural network approach adds a different level of security to applications of low-cost RFID tags.

3. Recharge day forecasting

The CLPCs are only as secure as their card manufacturer's security. This is not enough for applications that use many low-cost prepaid cards, such as those that are used in public transportation. If the vulnerability in the card security becomes apparent, the barriers to adopt a new secure card technology or to evolve a countermeasure are significant and time-consuming because software and hardware must be redesigned. A criminal organization can develop a business case for illicit exploitation of the system.

To overcome the potential security vulnerabilities, we propose a completely different security mechanism that depends on the cardholder's behavioral characteristics [17]. Individuals have behavioral characteristics that differentiate them from other people [23,24]. These distinct behavioral characteristics can be considered 'unique identifiers'. There are no rules that determine which behavioral characteristics define a unique identifier. Although there are many behavioral characteristics of a cardholder, we find three factors that serve as unique identifiers to characterize the relationship among daily use, the amount of recharge, and the elapsed day between recharges of the CLPC. This relationship is determined by trial-and-error method from 22,869 different cardholders during 3 years of use in public transportation systems in Turkey. With this relationship, we forecast the next recharge day and define an expiration date for a CLPC. Therefore, the cardholder must recharge his CLPC at a vending machine before the expiration date. Furthermore, we predict the cardholders' maximum daily use to provide a daily upper limit for use of the CLPC. With this security approach, it would be useless to clone the data of a CLPC.

4. Forecasting with neural networks

Forecasting is always subject to uncertainty from two sources: the model structure and the training data. It is necessary to identify an optimal combination from the set of input variables and the number of neurons in the hidden layer. Although there are different forecasting methods [25], the basic approach is to train a NN with historical data containing both inputs and the corresponding desired outputs. In this process, a NN constructs an input–output mapping and adjusts the weights and the biases at each iteration based on the minimization of an error measure between the produced and the desired outputs. The adequate selection of inputs, hidden

layers, training functions, tapped delay lines, and number of neurons strongly influences the success of the training process [26,27].

There are no guidelines for establishing an optimal configuration for an application. It is necessary to train different neural network models and choose the best option. Therefore, we use three types of multilayer perceptron (MLP)-based NN structures: the focused time delay neural network (FTDNN), the layer recurrent neural network (LRNN), and the nonlinear autoregressive neural network with external input (NARX). These MLPs have been used for many years in a wide range of applications, and there are many examples in the literature that explain the full mathematical treatment and structures of NNs and NN forecasting applications [19,28–31]. The MATLAB structure of the MLP used in this paper for forecasting the recharge time is given in the next section.

To identify the best forecasting result, different training functions and numbers of neurons are applied to the MLP. Each MLP consists of one input layer, one output layer, and a hidden layer with up to 30 neurons. The number of neurons in the hidden layer should be increased only when the results are not adequate. In our approach, one hidden layer produces excellent results, and there is no need to increase the size of the hidden layer. The tangent sigmoid function in the hidden layer and the linear function in the output layer are used as transfer functions. The following training functions are considered in the MLP: BFGS quasi-Newton (BFG), Bayesian regulation (BR), conjugate gradient with Powell/Beale restarts (CGB), Levenberg–Marquardt (LM), one-step secant (OSS), and scaled conjugate gradient (SCG).

5. Data analysis and training of the networks

The dataset used in this study comprises 3 years of data (2006–2008) for public transport with CLPCs. This dataset is obtained from 22,869 cardholders that belong to seven social groups (officials, workers, medics, high school students, undergraduate students, housewives, and retirees) in four different residential quarters in Turkey. The dataset consists of six input variables: the date, the day of the week, the time, the amount of daily use, the current balance per day, and the recharge amount. Figure 1 shows the current balance in Turkish lira (TL) per day of a CLPC used for public transportation by an official.

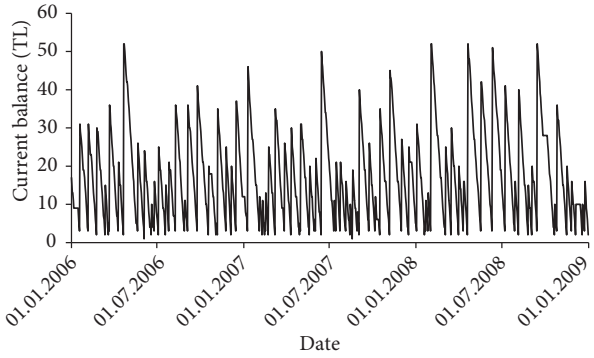

Figure 1. The current balance per day of a CLPC used for public transportation by an official (1097 data points).

The performance of a NN often deteriorates when the number of input variables increases. This has been referred to as the curse of dimensionality in the literature [26]. Increasing the number of input variables also leads to the need to use more training examples and process time to effectively understand the input-output relationship. Therefore, techniques such as principal component analysis or regression analysis are used to

decrease the number of input variables without losing any features from the dataset. However, we use the trial-and-error method to decrease the number of input variables because our dataset is comprehensive. We find that when we recalculate the elapsed time between recharges of the CLPC and use it with the recharge amount as input (Figure 2), the NN gives a very good forecast of the next recharge date. Therefore, three inputs (date of recharge, recharge amount, and elapsed days between the last recharge) and one output (forecast of the next recharge time) are used for every MLP.

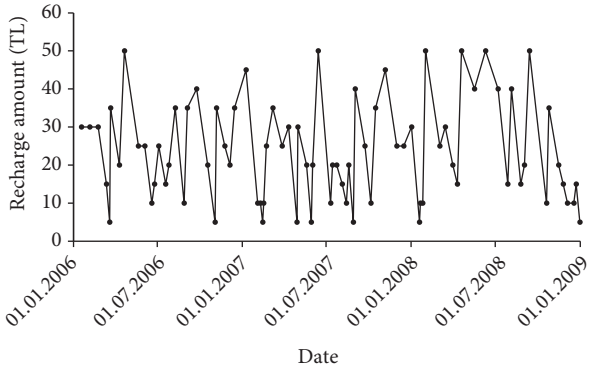

Figure 2. Recharge amount of a CLPC used in public transportation by an official (80 data points).

A large number of training processes are made with a NN, and two calculation methods are used to determine the most accurate forecast. The dataset is separated into two parts for the two different calculations. Figure 3 shows how the dataset is separated. All data points except the last data point (fresh data) are used for NN training. We investigate one-ahead forecasting, and, therefore, the last data point is evaluated separately. To implement the two calculation methods, the NN must be trained with the data. Therefore, data points used for NN training are divided into three parts: training (70%), validation (15%), and testing (15%). The dataset of 22,869 different cardholders involves approximately 80 data points for each cardholder, depending on their recharge times from January 2006 to December 2008. After the NN training, the resulting structure of the NN is used to predicting the training data and forecast the fresh data. These results are compared with the actual data points to determine the accuracy of the NN training.

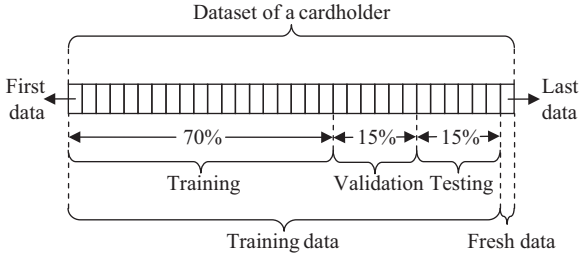

Figure 3. General format of the cardholder dataset used for NN training and one-ahead forecasting.

The first calculation method is the forecasting error analysis of one-ahead forecasting. The forecasting error analysis is calculated with four different functions. These functions are the relative error (RE), the forecast error (FE), the absolute error (AE), and the standard deviation (SD).

$$RE = \left| \frac{y - \hat{y}}{\hat{y}} \right| \cdot 100 \tag{1}$$

$$FE = y - \hat{y} \tag{2}$$

$$AE = |y - \hat{y}| \tag{3}$$

$$SD = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(FE - \overline{FE}\right)^2} \tag{4}$$

Here, y and $\hat{y}$ represent the actual and the forecasted (one-ahead forecasting or recharge day forecasting) values of the fresh data (Figure 3), respectively. $\bar{FE}$ indicates the average value of the one-ahead forecasting, and n represents the total number of cardholders. The variation of the one-ahead forecasting is calculated as $\bar{FE} \pm SD$. To obtain a highly accurate forecast of the recharge day, RE, FE, AE, and SD should be as small as possible.

The second calculation method is the prediction performance analysis of the NN training, and it is evaluated using the following functions: the root mean square error (RMSE), the mean absolute difference (MAD), the mean absolute percentage error (MAPE), and the Pearson correlation coefficient (r). Values of RMSE, MAD, and MAPE near 0 and r values near 1 show good performance by the NN training.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m} (x_i - \hat{x}_i)^{\mathbf{2}}}{m}} \tag{5}$$

$$MAD = \frac{\sum_{i=1}^{m} |x_i - \hat{x}_i|}{m} \tag{6}$$

$$MAPE = \frac{\sum_{i=1}^{m} \left| \frac{x_i - \hat{x}_i}{x_i} \right|}{m} \cdot 100 \tag{7}$$

$$r = \frac{\sum_{i=1}^{m} (x_i \cdot \hat{x}_i) - \frac{\sum_{i=1}^{m} x_i \cdot \sum_{i=1}^{m} \hat{x}_i}{m}}{\sqrt{\left(\sum_{i=1}^{m} x_i^2 - \frac{\left(\sum_{i=1}^{m} x_i\right)^2}{m}\right) \cdot \left(\sum_{i=1}^{m} \hat{x}_i^2 - \frac{\left(\sum_{i=1}^{m} \hat{x}_i\right)^2}{m}\right)}} \tag{8}$$

Here, x and $\hat{x}$ represent the actual and the predicted values, respectively, and m is the total number of data points in the training dataset (Figure 3).

In this research, each MLP is simulated with MATLAB's NN toolbox, and the best NN forecasting configuration is determined by varying the training functions (BFG, BR, CGB, LM, OSS, and SCG) and the number of neurons (S = 3–30). These MLPs are FTDNN, LRNN, and NARX with three inputs (date of recharge, recharge amount, and elapsed days since the last recharge) and one output (forecasting of the next recharge date). The best NN forecasting configuration is based upon minimizing the difference between the MLP forecasted value and the desired output. To avoid local minima far from the global minimum, each MLP is trained 20 times using different random initialization parameters. Therefore, the result for each neuron is the average of 20 different training processes. The training process of the MLP is stopped when either the default level of error or the default number of iterations is achieved.

6. Simulation results

In the present study, three basic steps are taken to determine the best MLP architecture due to the time-consuming MLP calculations, especially for the LRNN. In steps 1 and 2, the best architecture of each MLP is investigated. Finally, in step 3, these architectures are compared to define the best one.

First, to find the best training and prediction performance function, 100 cardholders are chosen randomly. Each training function is simulated 28 times for every cardholder because there are 28 neurons (S = 3–30) in the hidden layer. However, within these 28 results, only the best forecasting result is used, and the average from 100 cardholders is compared for every training function as shown in Tables 1–3. The best results are shown in bold. The smallest variation of the recharge day forecasting error ($\bar{FE} \pm SD$) is obtained with the LRNN and the SCG training function. However, if we compare it with the NARX and the LM training function, we see that the results are actually the same. Certainly, the best training function must be selected according to the smallest value of the prediction performance functions (the value of r that is closest to 1). In an actual recharge application, $\bar{RE}$, $\bar{AE}$, and $\bar{FE} \pm SD$ are unknown, and the one-ahead forecasting result is determined only from the smallest value of the prediction performance function. The training functions and their corresponding best prediction performance functions are summarized in Table 4. Furthermore, only the NARX and the FTDNN with the LM training function give the best forecasting error function results with two different prediction performance functions (Tables 1 and 3). Although none of the LRNN prediction performance functions give the best forecasting error function result, they are comparable to the FTDNN and the NARX. However, the average calculation time of the training process for each training function with the LRNN is extremely high (Table 5). Therefore, the LRNN could not be used in an actual recharge application (vending machine or recharge point), even if it had a better forecasting error function result. We use five computers with four different configurations (Intel i7 1.86 GHz, Intel i5 1.86 GHz, Intel Core 2 Duo 1.86 GHz, and two Intel Core 2 Duo 1.86 GHz PCs with 4.0 GB of RAM) for this and the following steps.

Table 1. The effect of different training functions on the recharge day forecasting performance for one-ahead forecasting with the FTDNN. Every result is an average of the data from 100 cardholders.

Training	Prediction performance functions				Forecasting error functions		
functions	$\overline{RMSE}$	$\overline{MAPE}$	$\overline{MAD}$	$\bar{r}$	$\overline{RE}$	$\overline{AE}$	$\overline{FE} \pm SD$
BFG	1.935	3.601	1.904	0.912	20.382	1.817	–0.490 ± 2.226
BR	3.819	5.677	1.785	0.789	17.762	1.715	–0.679 ± 1.941
CGB	**1.852**	3.465	**1.453**	0.917	21.793	1.810	–0.392 ± 2.233
LM	1.943	**3.290**	1.633	**0.923**	**14.931**	**1.423**	**–0.315** ± 1.875
OSS	2.188	5.189	1.775	0.895	25.588	1.916	–0.513 ± 2.466
SCG	1.980	4.437	1.507	0.911	21.385	1.797	–0.213 ± 2.267

Table 2. The effect of different training functions on the recharge day forecasting performance for one-ahead forecasting with the LRNN. Each result is the average of the data from 100 cardholders.

Training	Prediction performance functions				Forecasting error functions		
functions	$\overline{RMSE}$	$\overline{MAPE}$	$\overline{MAD}$	$\bar{r}$	$\overline{RE}$	$\overline{AE}$	$\overline{FE} \pm SD$
BFG	2.149	4.426	1.737	0.888	18.497	1.643	0.032 ± 2.105
BR	2.710	3.252	1.417	0.851	**12.023**	1.517	–0.290 ± 2.113
CGB	2.019	**2.935**	1.652	0.907	17.811	1.628	0.046 ± 2.034
LM	**1.702**	3.361	**1.309**	**0.931**	16.173	1.537	–0.016 ± 2.007
OSS	2.086	4.731	1.677	0.891	17.685	1.597	0.015 ± 1.982
SCG	2.013	3.519	1.564	0.898	16.641	**1.426**	**–0.044** ± 1.902

Table 3. The effect of different training functions on the recharge day forecasting performance for one-ahead forecasting with the NARX. Each result is the average of the data from 100 cardholders.

Training functions	Prediction performance functions				Forecasting error functions		
	$\overline{RMSE}$	$\overline{MAPE}$	$\overline{MAD}$	$\overline{r}$	$\overline{RE}$	$\overline{AE}$	$\overline{FE} \pm SD$
BFG	1.988	4.075	1.536	0.903	19.434	1.729	0.078 ± 2.250
BR	3.476	3.935	1.740	0.801	19.025	1.954	–0.793 ± 2.424
CGB	**1.856**	3.430	1.537	0.915	21.159	1.802	0.131 ± 2.371
LM	1.861	2.942	**1.383**	**0.926**	**14.844**	**1.467**	**0.117** ± 1.882
OSS	2.106	4.825	1.578	0.907	20.884	1.744	0.047 ± 2.440
SCG	1.991	**2.566**	1.599	0.910	20.372	1.781	–0.328 ± 2.212

Table 4. Training functions and their corresponding best prediction performance functions summarized from Tables 1–3.

MLP	Prediction performance functions			
	RMSE	MAPE	MAD	r
FTDNN	CGB	LM	CGB	LM
LRNN	LM	CGB	LM	LM
NARX	CGB	SCG	LM	LM

Table 5. Average calculation times in the second of the one-ahead forecasting processes for one cardholder.

Training functions	MLP		
	FTDNN	LRNN	NARX
BFG	650.426	10573.490	805.991
BR	3105.369	26105.086	588.869
CGB	113.365	1693.649	101.959
LM	11.249	497.235	10.901
OSS	144.439	1594.580	155.129
SCG	82.737	1847.193	110.628

Second, in order to determine the optimal number of neurons for the training function from step 1, another 100 cardholders are chosen randomly. After the forecasting error function, neurons that provide AE values below 3 are selected for each cardholder, and equal numbers of neurons from the 100 cardholders are grouped together. An upper limit of 3 for AE means a recharge day forecasting error within 3 days, which is good enough for use in public transportation applications. Furthermore, this entails the optimal range of neurons to make an acceptable selection for the next step. Increasing the number of neurons also increases AE, except for the LRNN with the CGB training function. Increasing the number of neurons in the hidden layer increases the power of the network at the expense of the computation time. Furthermore, it is more likely to produce over-fitting where the NN fits all of the training data but memorizes the training data instead of learning to adapt itself to new situations. Therefore, only neurons from Table 6 for the FTDNN, the LRNN, and the NARX are used in the next step, which provides better forecasting among the neurons. Table 7 shows the variation of the recharge day forecasting error performance for each MLP and its training function. The worst average calculation time for a cardholder is obtained again with the LRNN.

If we compare the RE and the AE (Table 7) with the values from the first step (Tables 1–3), only RE decreases significantly. Table 8 presents these decreases that depend on the number of elapsed days since the last recharge by the cardholder. According to these results in Table 8, only the AE is suitable for the accuracy of the recharge day forecasting.

Table 6. Top five neurons that provide AE values below 3 and the total number of neurons in the group (in brackets).

MLP	Training functions	Top five neurons				
FTDNN	CGB	3 (87)	4 (82)	5 (78)	8 (70)	7 (69)
	LM	4 (81)	5 (78)	8 (77)	3 (73)	6 (73)
LRNN	CGB	3 (76)	18 (74)	7 (73)	16 (72)	24 (72)
	LM	7 (81)	3 (79)	4 (77)	5 (76)	6 (74)
NARX	CGB	3 (81)	8 (77)	6 (74)	4 (73)	5 (81)
	LM	5 (81)	3 (80)	4 (75)	7 (75)	6 (73)
	SCG	4 (79)	3 (77)	8 (74)	5 (73)	6 (66)

Table 7. Recharge day forecasting error performance.

MLP	Training functions	$\overline{RE}$	$\overline{AE}$	$\overline{FE} \pm SD$	Time (s)
FTDNN	CGB	16.604	1.913	–0.085 ± 2.495	148.788
	LM	13.343	1.769	0.553 ± 2.153	11.170
LRNN	CGB	15.216	1.945	0.548 ± 2.311	1446.813
	LM	15.357	1.909	0.320 ± 2.381	367.407
NARX	CGB	17.095	2.052	0.102 ± 2.628	130.747
	LM	13.378	1.756	0.520 ± 2.171	22.508
	SCG	17.718	2.076	–0.122 ± 2.806	88.307

Table 8. Comparison of the RE and the AE, which refer to the number of elapsed days since the last recharge and the one-ahead forecasts for five cardholders.

Elapsed day between recharges	Forecasted value	RE	AE
4	4.538	13.454	0.538
11	9.528	13.384	1.472
12	10.396	13.366	1.604
19	16.486	13.230	2.514
21	18.198	13.343	2.802

In steps 1 and 2, the best architectures for each MLP are determined. Finally, to determine the best MLP, another 500 randomly chosen cardholders are used. These results are processed on the basis of the prediction performance functions as explained in step 1. Table 4 summarizes the results of step 1 and shows the best predictions achieved with the MLPs, the training functions, and the prediction performance functions. These results show, for example, that the FTDNN with the CGB training function gives good predictions with the RMSE (FTDNN-CGB-RMSE) or the MAD (FTDNN-CGB-MAD) prediction performance function. Furthermore, the FTDNN with the LM training function also gives good predictions when the MAPE (FTDNN-LM-MAPE) or the r (FTDNN-LM-r) prediction performance function is used. However, combinations of the FTDNN with the other training and prediction performance functions do not provide better prediction results. In this step, therefore, simulations are made with the FTDNN and the LRNN with two training functions and the NARX with three training functions. Therefore, every cardholder is simulated seven times. In step 2, these training functions and up to 30 neurons are investigated, and the best predictions are achieved with five neurons (Table 6). Therefore, there are 35 (7×5) different results for every cardholder, 10 with the FTDNN and the LRNN, and 15 with the NARX. For example, Table 9 shows the NARX results for a cardholder. This table is shown as an Excel sheet to explain step 3. These results are obtained with the CGB-the RMSE (rows 2–6), the SCG-MAPE (rows 7–11), the LM-MAD (rows 12–16), and the LM-r (rows 12–16). However, to compare the

MLP and the training functions, the best value from every prediction performance function is used to determine the best MLP. The smallest RMSE value in rows 2–6 (NARX-CGB-RMSE) is in cell B-4, and row 4 is selected. For the MAPE, the smallest value in rows 7–11 (NARX-SCG-MAPE) is in cell C-11, and row 11 is selected. For the MAD, the smallest value in rows 12–16 (NARX-LM-MAD) is in cell D-12, and row 12 is taken. Finally, for r, the highest value in rows 12–16 (NARX-LM-r) is in cell E-13, and row 13 is taken. Thus, there are only four results for every cardholder, each one obtained from different prediction performance functions' best value. These procedures are repeated for every 500 cardholders, and the average values are shown in Table 10.

Table 9. Simulation results of the NARX-CGB-RMSE (rows 26), the NARX-SCG-MAPE (rows 711), the NARX-LM-MAD, and the NARX-LM-r (rows 1216).

	A	B	C	D	E	F	G
1	Neuron	RMSE	MAPE	MAD	r	Actual value	Forecasted value
2	3	1.3930	1.9669	1.0881	0.9776	21	20.4798
3	4	1.4887	1.3722	1.1694	0.9737	21	20.1594
4	5	1.3449	1.2137	1.0939	0.9790	21	20.2655
5	6	1.4190	2.3633	1.0798	0.9758	21	20.3924
6	8	1.4047	1.7773	1.1194	0.9770	21	19.7813
7	3	1.4837	2.5216	1.1574	0.9747	21	20.5816
8	4	1.4753	2.4746	1.2053	0.9746	21	19.9875
9	5	1.4698	1.8973	1.1513	0.9746	21	19.9637
10	6	1.4323	1.7303	1.1391	0.9759	21	20.1742
11	8	1.4886	1.1931	1.1693	0.9738	21	20.0760
12	3	1.2602	2.5542	0.9591	0.9836	21	20.4614
13	4	1.3962	1.7698	1.0418	0.9837	21	20.1785
14	5	1.3565	2.9720	1.0420	0.9818	21	20.6051
15	6	1.2924	1.7560	0.9681	0.9826	21	20.6338
16	7	1.3375	0.2655	2.0491	0.9827	21	20.1186

The NARX with the LM training function and the MAD prediction performance function gives the best one-ahead forecast, and it is also the fastest technique, whereas LRNN is again the slowest method. The subsequent results presented in this paper are obtained using the NARX with the LM training function and the MAD prediction performance function. We use 38 computers with two different configurations (AMD Phenom 2.30 GHz and Intel Core 2 Duo 3.00 GHz PCs with 2.0 GB of RAM) for this step and the following section.

Table 10. Comparisons of the recharge day forecasting error performance.

MLP	Training functions	Performance functions	$\overline{AE}$	$\overline{FE} \pm SD$	Time (s)
FTDNN	CGB	RMSE	**1.926**	**–0.354 ± 2.466**	13.232
	LM	MAPE	1.970	–0.187 ± 2.808	3.780
	CGB	MAD	1.934	–0.375 ± 2.498	13.232
	LM	r	2.002	–0.155 ± 2.900	3.780
LRNN	LM	RMSE	1.936	–0.142 ± 2.542	107.652
	CGB	MAPE	1.930	–0.431 ± 2.516	312.460
	LM	MAD	1.948	–0.165 ± 2.535	107.652
	LM	r	**1.928**	**–0.130 ± 2.542**	107.652
NARX	CGB	RMSE	1.910	–0.416 ± 2.468	12.105
	SCG	MAPE	1.991	–0.486 ± 2.584	7.945
	LM	MAD	**1.868**	**–0.106 ± 2.411**	3.893
	LM	r	1.933	–0.128 ± 2.592	3.893

After determining the best MLP and its training function, its prediction performance function, and the optimum number of neurons, the remaining 22,169 cardholders are investigated and validated. Table 11 shows that for over 90% of the cardholders, the recharged day forecasting error is less than 3 days. Detailed distributions of the recharge day forecasting error with occupations are shown in Table 12. These results show that our approach is capable of successfully forecasting the uncertain irregular recharge days of cardholders, such as housewives, undergraduate students, and retirees.

Table 11. Distribution of the recharge day forecasting errors with the MLP NARX, the LM training function, and the MAD prediction performance function.

	Number of cardholders	Percentage of cardholders
AE ≤ 1	8976	40.489
$1.0 <$ AE ≤ 1.5	4023	18.147
$1.5 <$ AE ≤ 2.0	3133	14.132
$2.0 <$ AE ≤ 2.5	2526	11.394
$2.5 <$ AE ≤ 3.0	1462	6.595
$3.0 <$ AE ≤ 3.5	1014	4.574
$3.5 <$ AE ≤ 4.0	563	2.540
$4.0 <$ AE ≤ 4.5	341	1.538
$4.5 <$ AE ≤ 5.0	129	0.582
AE > 5	2	0.009

Table 12. Detailed distributions of the recharge day forecasting errors in descending order for AE ≥ 1.

Occupation	Number of cardholders	Percentage of cardholders		
		AE ≤ 1	$1 <$ AE ≤ 3	AE > 3
High school student	1916	54.611	44.053	1.336
Medic	3539	49.091	46.654	4.255
Worker	5048	40.865	52.611	6.524
Official	6541	38.201	51.075	10.724
Retirees	1171	35.095	49.358	15.547
Undergraduate student	1470	33.199	50.179	16.623
Housewife	2484	29.283	51.575	19.142

7. Conclusions and future work

We propose a novel NN forecasting approach to prevent counterfeiting and cloning for closed-loop prepaid cards based on RFID or 1-Wire technology without an online backend system. Other researchers have shown that it is possible to clone or emulate high-security cards with 3DES or AES cryptographic protocols and factory-programmed UID. Furthermore, other researchers have shown that there are security weaknesses in online backend systems with high security cards. The main reason for these weaknesses is their security concept, where the security key and the UID are individual and specific but nonchangeable for each card. The whole security protocol is the same for every user. Although our proposed security approach does not modify or improve the existing protocol of the RFID system, it proposes another level of security that is individual, specific, and dynamic for each user. This is archived by defining an expiration date for each card according to the cardholders' recharge day and behavioral characteristics with NN forecasting. Simulation results show that this evolved NN forecasting configuration can successfully predict the next recharge time. For over 90% of the cardholders, the recharge days are forecast with a deviation of less than 3 days, which is a great performance for use in public transportation systems.

In this paper, we show that the behavioral characteristics of cardholders can be used with NN forecasting to define security levels that are individual, personal, and dynamic. Therefore, the implementation of our proposed security approach in existing RFID applications with low-cost tags is suggested. With our security approach, cloning a card would be useless, and in the worst-case scenario, a cloned card could only be used until its expiration date. Although a simple maximum daily use of a card is defined (the recharge amount divided by the predicted next recharge time), a more precise definition of the maximum daily use for every weekday should be investigated with NN forecasting as a second behavior characteristic of cardholders as a future work. If the upper limit of daily use is reached, then the cardholder must show his/her card to the bus driver before they can use public transportation. Genuine cards bear holograms or stamps from the public transport authority. Therefore, cloned cards could only be used once a day until the expiration date. In future work, other behavior characteristics of cardholders should be investigated to prevent the daily use until the expiration date.

Acknowledgment

This research was supported as part of the project "Campus RFID Automation System" at Sakarya University in Turkey.

References

[1] Nohl K, Plöz H. Mifare, little security, despite obscurity. In: Congress of the Chaos Computer Club; 27–30 December 2007; Berlin, Germany.

[2] Nohl K, Evans D, Starbug S, Plotz H. Reverse-engineering a cryptographic RFID tag. In: USENIX Security Symposium; 28 July–1 August 2008; San Jose, CA, USA.

[3] Courtois NT, O'Neil S, Quisquater JJ. Practical algebraic attacks on the Hitag2 stream cipher. In: Information Security Conference; 7–9 September 2009; Pisa, Italy. pp. 167–176.

[4] de Koning Gans G, Hoepman JH, Garcia FD. A practical attack on the MIFARE Classic. In: Smart Card Research and Advanced Application Conference; 8–11 September 2008; London, UK. pp. 267–282.

[5] Mayes KE, Cid C. The MIFARE Classic story. Information Security Technical Report 2010; 15: 8–12.

[6] Garcia FD, de Koning Gans G, Muijrers R, van Rossum P, Verdult R, Schreur RW, Jacobs B. Dismantling MIFARE Classic. In: European Symposium on Research in Computer Security; 6–8 October 2008; Malaga, Spain. pp. 97–114.

[7] Courtois NT. The Dark side of security by obscurity and cloning MiFare Classic rail and building passes anywhere, anytime. In: International Conference on Security and Cryptography; 7–10 July 2009; Milan, Italy. pp. 331–338.

[8] Teepe W. Making the Best of Mifare Classic. Nijmegen, the Netherlands: Radboud University, 2008.

[9] Garcia FD, van Rossum P, Verdult R, Schreur, RW. Wirelessly pickpocketing a Mifare Classic card. In: IEEE Symposium on Security and Privacy; 17–20 May 2009; Berkeley, CA, USA. pp. 3–15.

[10] Grand J. Protecting your crown jewels: an introduction to embedded security for hardware-based products. Comput Fraud Secur 2005; 10: 13–20.

[11] Sanghera P, Thornton F, Haines B, Kleinschidt J, Das AM, Bhargava H, Campbell A. How to Cheat at Deploying and Securing RFID. Burlington, MA, USA: Syngress, 2007.

[12] Bono SC, Green M, Stubblefield A, Juels A, Rubin AD, Szydlo M. Security analysis of a cryptographically-enabled RFID device. In: USENIX Security Symposium; 31 July–5 August 2005; Baltimore, MA, USA. pp. 1–16.

[13] Russell R, Kaminsky D, Puppy RF, Grand J, Ahmad D, Flynn H, Dubrawsky I, Manzuik SW, Permeh R. Hack Proofing Your Network. 2nd ed. Rockland, MA, USA: Syngress, 2002.

[14] Kasper T, Oswald D, Paar C. EM side-channel attacks on commercial contactless smartcards using low-cost equipment. Lect Notes Comput Sc 2009; 5932: 79–93.

[15] Kasper T, von Maurich I, Oswald D, Paar C. Cloning cryptographic RFID cards for 25$. In: Benelux Workshop on Information and System Security; 29–30 November 2010; Nijmegen, the Netherlands.

[16] Murdoch SJ, Drimer S, Anderson R, Bond M. Chip and PIN is broken. In: IEEE Symposium on Security and Privacy; 16–19 May 2010; Berkeley, CA, USA. pp. 433–446.

[17] Jain AK, Ross A, Prabhakar S. An introduction to biometric recognition. IEEE T Circ Syst Vid 2004; 14: 4–20.

[18] Iranmanesh SH, Zarezadeh M. Application of artificial neural network to forecast actual cost of a project to improve earned value management system. World Academy of Science Engineering and Technology 2008; 42: 210–213.

[19] Xie H, Tang H, Liao YH. Time series prediction based on Narx neural networks: an advanced approach. In: International Conference on Machine Learning and Cybernetics; 12–15 July 2009; Baoding, Hebei, China. pp. 1275–1279.

[20] Pilka F, Oravec M. Multi-step ahead prediction using neural networks. In: International Symposium ELMAR; 14–16 September 2011; Zadar, Croatia. pp. 269–272.

[21] Khosravi A, Nahavandi S, Creighton D. Quantifying uncertainties of neural network-based electricity price forecasts. Appl Energ 2013; 112: 120–129.

[22] Svalina I, Galzina V, Lujic R, Simunovic G. An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: the case of close price indices. Expert Syst Appl 2013; 40: 6055–6063.

[23] Henniger O, Nikolov D. Extending EMV payment smart cards with biometric on-card verification. Int Fed Info Proc 2013; 396: 121–130.

[24] Deutschmann I, Nilsson L, Nordstrom P. Continuous authentication, using behavioral biometrics, with keystroke and mouse. IT Professional 2013; 99: 1–4.

[25] Li Z, Rose JM, Hensher DA. Forecasting automobile petrol demand in Australia: an evaluation of empirical models. Transport Res A-Pol 2010; 44: 16–38.

[26] Mazloumi E, Rose G, Currie G, Moridpour S. Prediction intervals to account for uncertainties in neural network predictions: methodology and application in bus travel time prediction. Eng Appl Artif Intel 2011; 24: 534–542.

[27] Huang W, Lai KK, Nakamori Y, Wang S, Yu L. Neural networks in finance and economics forecasting. Int J Inf Tech Decis 2007; 6: 113–140.

[28] Swingler K. Applying Neural Networks: A Practical Guide. San Francisco, CA, USA: Morgan Kaufmann, 1996.

[29] Heaton J. Introduction to the Math of Neural Networks (Beta-1, e-Book). Chesterfield, MO, USA: Heaton Research, 2011.

[30] Florita AR, Henze GP. Comparison of short-term weather forecasting models for model predictive control. HVAC&R Res 2009; 5: 835–853.

[31] Medsker LR, Jain LC. Recurrent Neural Networks Design and Applications. Boca Raton, FL, USA: CRC Press, 2001.