Functional Electrical Stimulation System for a Wearable-based Biostimulation

João Lucas Gonçalves[1], José Lima[1], and Paulo Leitao[1]

Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, Portugal
a37887@alunos.ipb.pt,{jllima,pleitao}@ipb.pt

Abstract. The objective of this paper is to describe the methodology and preliminary results of the development of an electrical stimulation system, for a wearable device aiming the biostimulation. The paper describes the project environment of this work, the function and importance of the functional electrical stimulation, the current development of the system and presents the preliminary results with further expectations.

Keywords: Functional Electrical Stimulation · Wearable · Stimulation circuit.

1 Introduction

The Vastus Medialis muscle is a part of the quadriceps muscle group [3] [6]. The weakness in this muscle is one of the main factors that could provoke some patellar illness [5] [2]. This weakness may be caused by muscle atrophies, and in order to prevent those kinds of situations, physical exercise is essential. However, in a few cases, only the exercises may not be enough to provide the strength needed to properly treat them. Seen this, the Functional Electrical Stimulation (FES) become a good option in the reinforcement of the muscle, proving to be one of the best solutions to help on the treatment.

Between the existing solutions on the market, most of them have clear disadvantages compared with the proposed solution. The commercial solutions can provide the waveform required for the treatment in a "closed" matter, by applying a signal with constant parameters defined by levels of operation. In other words, the systems have a routine with preprogrammed stimulation patterns, which implements a constant value for the impulse parameters based on a lookup table [1]. Furthermore, those equipment's are bulky or even static, which restricts the movement capacity of the patients.

The main challenges of this project are to develop a system with an open architecture to apply a required pulse for which situation, and make it small enough to fit in a wearable device with the proper autonomy and power to fulfill the required parameters for the treatment.

Having this in mind, the objective of this work is to develop a system capable of receiving an impulse signal determined by the control system according to the therapy procedure and apply this impulse without any kind of harm to the patient using the electrodes embedded in a wearable device. This paper describes the ongoing work regarding the development of an electrical stimulation system to be incorporated in such wearable-based biostimulation system, and analyses the preliminary results.

The rest of the paper is organized as follows. Section 2 overviews the architecture for the wearable-based integrated biostimulation system, particularly including the signal acquisition and conditioning system and the functional electrical stimulation. Section 3 presents the functional electrical stimulation system, and Section 4 presents the preliminary results. Finally, Section 5 rounds up the paper with the conclusions and points out the future work.

2 Wearable-based Integrated Biostimulation System

NanoStim is a development project of a wearable device to assist in the treatment of knee pathology's caused by a deficiency on the vastus medialis muscle. The project is subdivided into three parts, those are acquisition, conditioning system, and electrical stimulation.

The first part aims to acquire the Electromyography (EMG) signal generated by the muscle, the second one analyzes the data using a machine learning system, and answer with a signal to the third part, who generates an electrical stimulus to assist the muscle activities. The basic architecture can be observed in Fig. 1.

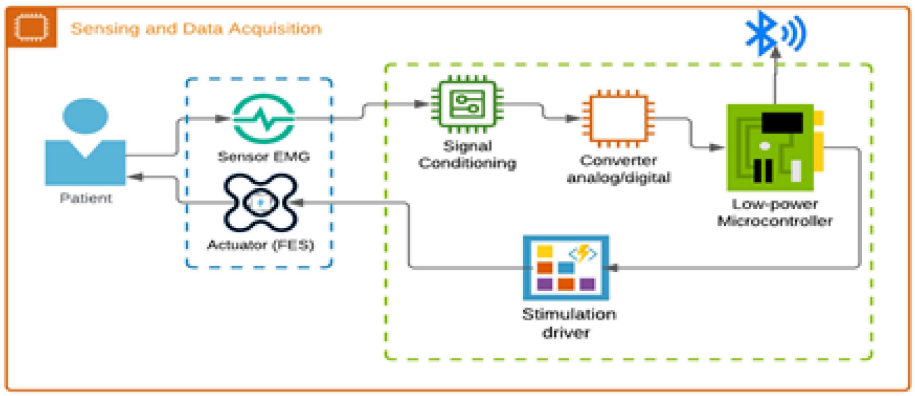

Fig. 1: Architecture of the acquisition, signal processing and electrostimulation system.

3 FES System Architecture

Seen this, the goal of this part of the project is to develop a system capable of receiving a command signal by wireless communication and generate an electrical stimulus. For this, is necessary to comprehend the concept of electrical stimulation, understand the functioning of the electrodes and their connections, and develop a circuit capable of delivering the appropriate current signal.

3.1 Functional Electrical Stimulation

Functional electrical stimulation generates muscle contractions by artificially inducing a current in specific motor neurons. Those electrical currents are delivered in pulses

using electrodes. These electrodes can be wet or dry and can be placed on the skin surface, within a muscle, on the surface of the muscle, or around the muscle nerve [4].

The tension produced on the electrically stimulated muscle depends on the intensity and frequency of the stimulation. This intensity is determined by a function of the charge transferred to the muscle, which depends on the amplitude, duration, frequency ,and shape of the pulse [4]. A typical electrical stimulation pulse is shown in figure 2.

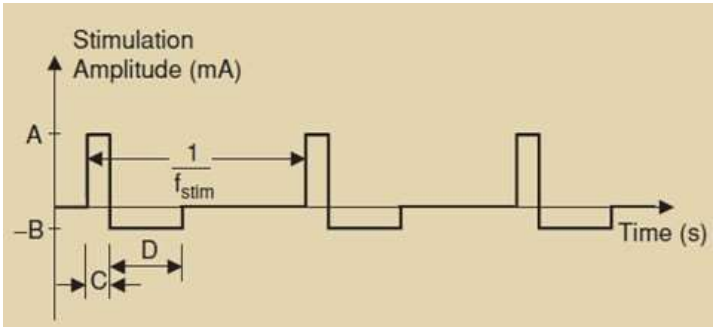

Fig. 2: Electrical stimulation pulse. A typical waveform, a square-wave pulse train used for transcutaneous FES [4].

3.2 The Stimulation Circuit

The stimulation circuit needs to receive a command and generates a current pulse with the pre-determined parameters. So the circuit is divided into two parts, software ,and hardware.

The preliminary hardware circuit is composed of an integrated circuit (IC) that sends impulses to a circuit composed of a pair of transistors and discreet components. Those impulses will be translated to the appropriate waveform by the components, and the output voltage has to be amplified to achieve the necessary voltage to stimulate the muscles.

To fulfill the amplification of the output voltage was studied two possibilities, the use of a low power elevator transformer and a DC/DC step-up converter. During the test was verified that the required voltage for stimulation was around 45 to 48 volts, so the step-up converter was discarded once it didn't provide the required output value. Seen this, the transformer is the best alternative.

The software part is the one that receives the command and provides the impulses with the specific characteristics for the stimulation signal. So it must be configured to produce the impulse with the values of the amplitude, frequency, duration ,and shape acquired by the IC.

It is important to consider that the circuit must have an open architecture and low power consumption once will be integrating into a wearable device so it can not be bulky, and needs to adapt to distinguished stimulation patterns.

4 Preliminary Results

The circuit was tested using pre-determined values, once the software part is not ready to receive the acquisition parameters. In this way, the circuit receives two impulses of 100μ seconds, one after the other, every 20m seconds, and the amplitude of the pulses was regulated by an external power source at 3V. The period and waveform of the signal can be observed in figures 3 and 4.

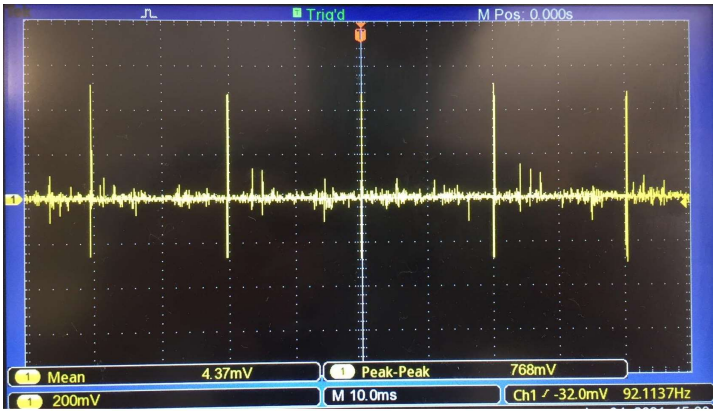

Fig. 3: Impulse period

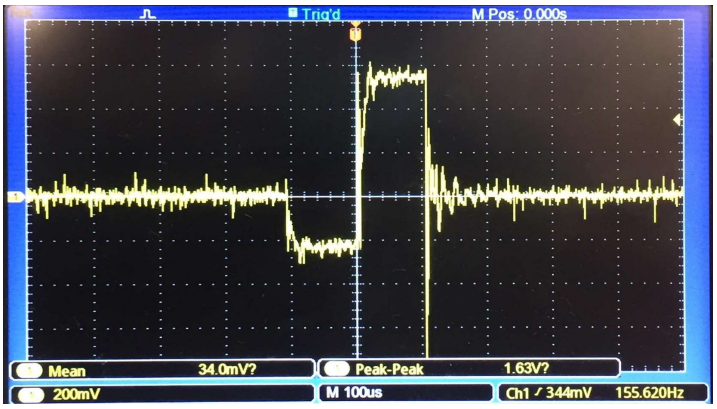

Fig. 4: Wave form

Figure 3 shows the impulses received by the circuit every 20m seconds, and figure 4 presents the waveform obtained with the determined value of 100μ seconds for the pulse duration.

5 Conclusions

The preliminary results show that the designed circuit is able to generate the estimated stimulation pattern, within the proposed period determined by the control module. The obtained waveform and period of the signal constitute the base components to set up the proper output signal to fulfill the stimulation.

However, the output value is inferior to the one needed to fulfill the stimulation, which requires further development to reach the appropriate output voltage for the system. Additionally, future work will be devoted to improving the quality of the waveform, and integrating the FES system with the rest of the data acquisition and processing system.

Acknowledgment

The work reported in this paper was supported by Nanostim - Nanomaterials for wearable-based integrated biostimulation, project n. POCI-01-0247-FEDER-045908 (co-financed by COMPETE 2020, Portugal 2020, Fundos Europeus e Estruturais e de Investimento).

References

1. Cheng, K.E., Lu, Y., Tong, K.Y., Rad, A., Chow, D.H., Sutanto, D.: Development of a circuit for functional electrical stimulation. IEEE Transactions on neural systems and rehabilitation engineering **12**(1), 43–47 (2004)
2. Grob, K., Gilbey, H., Manestar, M., Ackland, T., Kuster, M.S.: The anatomy of the articularis genus muscle and its relation to the extensor apparatus of the knee. JBJS Open Access **2**(4) (2017)
3. Grob, K., Manestar, M., Filgueira, L., Kuster, M.S., Gilbey, H., Ackland, T.: The interaction between the vastus medialis and vastus intermedius and its influence on the extensor apparatus of the knee joint. Knee Surgery, Sports Traumatology, Arthroscopy **26**(3), 727–738 (2018)
4. Lynch, C.L., Popovic, M.R.: Functional electrical stimulation. IEEE control systems magazine **28**(2), 40–50 (2008)
5. Sakai, N., Luo, Z.P., Rand, J.A., An, K.N.: The influence of weakness in the vastus medialis oblique muscle on the patellofemoral joint: an in vitro biomechanical study. Clinical Biomechanics **15**(5), 335–339 (2000)
6. Sandercock, T.G., Wei, Q., Dhaher, Y.Y., Pai, D.K., Tresch, M.C.: Vastus lateralis and vastus medialis produce distinct mediolateral forces on the patella but similar forces on the tibia in the rat. Journal of biomechanics **81**, 45–51 (2018)

DNS firewall based on Machine Learning: Proposal, Methodology and Preliminary Results

Cláudio Marques[1], Silvestre Malta[2], and João Paulo Magalhães[3]

[1] Instituto Politécnico de Viana do Castelo, IPVC, Portugal
[2] ADiT-Lab, Instituto Politécnico de Viana do Castelo, IPVC, Portugal
[3] CIICESI, Instituto Politécnico do Porto, IPP, Portugal
claudioms@ipvc.pt,smalta@estg.ipvc.pt,jpm@estg.ipp.pt

Abstract. In this paper we present a data analytic process that involves the creation of a Domain Name Service (DNS) dataset to be trained by machine learning algorithms in order to detect malicious DNS domains on the fly. The dataset is based on real DNS logs and it was enriched using Open Source Intelligence (OSINT) sources. The exploratory analysis and data preparations steps were carried and the final dataset will be submitted to different Machine Learning (ML) algorithms. Some preliminary results reveals the accuracy and time required to classify if a domain request is malicious or not.

Keywords: Cybersecurity · DNS · Firewall · Machine Learning.

1 Introduction

The DNS is a fundamental service to the functioning of the Internet. As the Internet grows, so does the number of DNS domains. According to the report provided by Verisign [7], the second quarter of 2020 closed with 370.1 million domain name registrations across all Top Level Domain (TLD). An increase of 0.9 percent when compared to the first quarter of 2020 and a growth of 4.3 percent, year over year. Unfortunately, this growth has a less positive side. In [6] it is said that 70% of newly 200,000 registered domains every day are malicious or suspicious.

In this paper we present a proposal to build a DNS firewall based-on machine learning. The proposal includes the creation of a dataset from scratch and the posterior data preparation and analysis in order to assess its accuracy. The dataset contains 90000 different domains classified *a priori* as malicious or benign. From the DNS query were derived 34 different features. The missing values, outlier analysis and data distribution were analyzed and the resulting dataset submitted to different machine learning supervised classification algorithms.

The rest of the paper is organized as follows: Section 2 presents the related work; the proposal and methodology is presented in section 3; the data analytics process and the preliminary results are presented in section 4; section 5 concludes the paper.

2 Related Work

Several studies investigate the security of DNS systems and how to improve the detection of cyber threats using ML algorithms. In [2] authors used nine features of botnet domain querying and a popular classifier algorithm to pick the malicious domains out

of DNS traffic. The results obtained using the Random Forest classifier algorithm reach the 99.38%, a false positive rate of 0.28% and a false negative rate of 1.86%. A more recent study presented in [3] combines blocklists / allowlists with a ML approach on DNS traffic.A deep neural architecture model was trained using passive DNS database. This study was able to detect if a domain is benign, malign or a sinkhole with 95% of accuracy on malicious and a false positive rate of 1:1000. The study also uses different algorithms and presents a comparison table for each one. To avoid the take-down of botnets by the law enforcement authorities, malicious authors make use of Domain Generation Algorithm (DGA). This technique generates random domain names that change over time, making the sinkhole process and botnet take down difficult. In [4], a ML model is proposed to identify DGA domain names and improve the security of DNS firewall solutions.

The use of DNS Response Policy Zone (RPZ) to block malicious domains is still a useful approach [8]. Using ML algorithms, such the logistic regression classification algorithm, to actively identify possible threats at DNS level as presented in [5] combined with a DNS RPZ approach can be valuable and improve the detection of malicious domains.

3 Proposal and Methodology

Considering the number of newly domains and the growing number of domains associated with malicious activities, the development of a DNS firewall is of utmost importance. It is meant to protect users from accessing malicious domains, preventing the installation of malware, the communication with command and control servers, the access to phishing websites and data exfiltration.

A system as we propose can be used to complement the DNS firewall systems based on DNS block and allow lists with algorithms capable of verifying if a domain is malicious or not. In the first phase, a dataset was created using non-malicious and malicious domains. The second phase focus on the preliminary results achieved by different ML applied on the dataset. This phase is still in progress, so some results could improve or change over time as we study and fine tune the parameters and hyper parameter for each algorithm.

4 Data Analytics

In this section we present the data analysis process. This process start with the creation of the dataset, the posterior data preparation and then the data analysis phase that allows us to obtain the first classification results regarding if a given domain is malicious or not.

4.1 Creating the dataset

To create the dataset we started from lists of already classified malicious and non-malicious domains (data sources). All started with a simple domain name and for each domain a DNS query was performed. The results were logged and then processed.

Several Python modules were created to extract the different features from the domain name. So, using the domain name as input, 34 features (described in Table 1) were obtained.

Table 1: Dataset features with description, data types and default value

Feature	Description	Data Type					Default Value
		Text	Boolean	Integer	Decimal	Enumerate	
Domain	Baseline DNS used to enrich data (derive features)	X					N/A
DNSRecordType	DNS record type queried	X					N/A
MXDnsResponse	The response from a DNS request for the record type MX		X				False
TXTDnsResponse	The response from a DNS request for the record type TXT		X				False
HasSPFInfo	If the DNS response has Sender Policy Framework attribute		X				False
HasDkimInfo	If the DNS response has Domain Keys Identified Email attribute		X				False
HasDmarcInfo	If the DNS response has Domain-Based Message Authentication		X				False
Internet Protocol (IP)	The IP for the domain	X					null
DomainInAlexaDB	If the domain it's registered in the Alexa DB		X				False
CommonPorts	If the domain it's available for common ports (80, 443, 21, 22, 23, 25, 53, 110, 143, 161, 445, 465, 587, 993, 995, 3306, 3389, 7547, 8080, 8888)		X				False
CountryCode	The country code associated with the IP of the domain	X					null
RegisteredCountryCode	The country code defined in the domain registration process (WHOIS)	X					null
CreationDate	The creation date of the domain (WHOIS)					X	0
LastUpdateDate	The last update date of the domain (WHOIS)					X	0
ASN	The Autonomous System Number for the domain			X			-1
HttpResponseCode	The HTTP/HTTPS response code for the domain					X	0
RegisteredOrg	The organization name associated with the domain (WHOIS)	X					null
SubdomainNumber	The number of sub-domains for the domain			X			0
Entropy	The Shannon Entropy of the domain name			X			0
EntropyOfSubDomains	The mean value of the entropy for the sub-domains			X			0
StrangeCharacters	The number of characters different from [a-zA-Z] and considering the existence maximum of two numeric integer values			X			0
TLD	The Top Level Domain for the domain	X					null
IpReputation	The result of the blocklisted search for the IP		X				False
DomainReputation	The result of the blocklisted search for the domain		X				False
ConsoantRatio	The ratio of consonant characters in the domain				X		0
NumericRatio	The ratio of numeric characters in the domain				X		0
SpecialCharRatio	The ratio of special characters in the domain				X		0
VowelRatio	The ratio of vowel characters in the domain				X		0
ConsoantSequence	The maximum number of consecutive consonants in the domain			X			0
VowelSequence	The maximum number of consecutive vowels in the domain			X			0
NumericSequence	The maximum number of consecutive numerics in the domain			X			0
SpecialCharSequence	The maximum number of consecutive special characters in the domain			X			0
DomainLength	The length of the domain			X			N/A
Class	The class of the domain (malicious = 0 and non-malicious = 1)			X			N/A

Features like the domain name entropy, number of strange characters and domain name length were obtained directly from the domain name. Other features like, domain name creation date, IP, open ports, geolocation were obtained from data enrichment processes (e.g. OSINT). The class was determined considering the data source (malicious DNS log files and non-malicious DNS log files). The dataset consists of data from approximately 90000 domain names and it is balanced between 50% non-malicious and 50% of malicious domain names.

4.2 Data preparation, Data analysis and Preliminary results

On data preparation the Domain and Ip columns were anonymized on the dataset. For the text type columns we use the Label Encoder from SkLearn framework [1] with a value between 0 and $n\_classes - 1$. The boolean type was transformed to integers (0 and 1). For all integers it was used the min-max normalization exporting all the values to a range between 0 and 1. The algorithms used to the preliminary evaluation

were: Support-Vector Machine (SVM), Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Decision Tree (CART) and Naive Bayes (NB). Two different algorithms were used to select the most significant features for the analysis: (1) Extra Trees algorithm and (2) Univariate feature selection. In both cases, the dataset was split between training and testing groups using 10-fold cross validation. For the number of folds (10), the analysis of the state of the art was taken into account. The results are presented in Table 2.

Table 2: Table with Machine Learning results using Extra Trees

Algorithm	**Test Mode**	**Feature selection (1)**	**Accuracy**	**Time (sec)**	**Feature selection (2)**	**Accuracy**	**Time (sec)**
SVM	10-fold-cross-validation	9 features	0.912456	499.165	9 features	0.919911	534.308
LR	10-fold-cross-validation	9 features	0.916622	5.2711	9 features	0.916711	4.03559
LDA	10-fold-cross-validation	9 features	0.908822	1.19985	9 features	0.906233	1.18046
KNN	10-fold-cross-validation	9 features	0.949789	47.456	9 features	0.947333	79.0447
CART	10-fold-cross-validation	9 features	0.947833	0.950081	9 features	0.956022	0.774204
NB	10-fold-cross-validation	9 features	0.903156	0.46814	9 features	0.908289	0.359953

The results obtained so far are encouraging. These results show that the algorithms are able to distinguish between benign and malicious domains with accuracy rates between 90% and 95%. Considering that the time to detect malicious domains is very important, the analysis presented also considers the time, in seconds, that each algorithm took to make the decision.

5 Conclusion

In this paper we presented a proposal for the creation of a machine learning-based DNS firewall solution. This work is an important contribution to the improvement of the DNS firewall legacy systems based on block/allow lists depending on previous known malicious domains. It is possible to apply the dataset on ML algorithms and, depending on the accuracy and execution time, select the best approach and implement a real-time DNS firewall alert system in order to increase the security of the internet usage. The preliminary results on this dataset and the correlation of various ML algorithms allow us to have a general idea of the most accurate, precise and timeless algorithm to study in advance. Since the results are mainly above 90% we expected that with the application of techniques on the ML algorithms, such hyper parameter tuning, it will be possible to build a significant model to be applied in a real scenario. The public availability of the created dataset to the scientific community is also important, since it allows the methodology to be adopted for the implementation of a DNS firewall that will take advantage of Supervised Classification Machine Learning Algorithms to detect if a given domain is potentially malicious or not.

References

1. Scikit Learn (2020), `https://scikit-learn.org/stable/`.

2. Guo, Z., Peng, J., Fu, J., Cheng, Y., Chen, C.: Botnet detection method based on artificial intelligence. In: Proceedings - 2019 IEEE 4th International Conference on Data Science in Cyberspace, DSC 2019. pp. 487–494. Institute of Electrical and Electronics Engineers Inc. (jun 2019). https://doi.org/10.1109/DSC.2019.00080
3. Lison, P., Mavroeidis, V.: Neural reputation models learned from passive DNS data. In: Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017. vol. 2018-Janua, pp. 3662–3671. Institute of Electrical and Electronics Engineers Inc. (jul 2017). https://doi.org/10.1109/BigData.2017.8258361
4. Mao, J., Zhang, J., Tang, Z., Gu, Z.: DNS anti-attack machine learning model for DGA domain name detection. Physical Communication **40**, 101069 (jun 2020). https://doi.org/10.1016/j.phycom.2020.101069
5. Palaniappan, G., Sangeetha, S., Rajendran, B., Sanjay, Goyal, S., Bindhumadhava, B.S.: Malicious Domain Detection Using Machine Learning on Domain Name Features, Host-Based Features and Web-Based Features. In: Procedia Computer Science. vol. 171, pp. 654–661. Elsevier B.V. (jan 2020). https://doi.org/10.1016/j.procs.2020.04.071
6. scmagazine: Vast majority of newly registered domains are malicious (2019), `https://www.scmagazine.com/home/security-news/malware/vast-majority-of-newly-registered-domains-are-malicious`
7. Verisign: The domain name industry brief (2020), volume 17, Issue 3
8. Wilde, N., Jones, L., Lopez, R., Vaughn, T.: A DNS RPZ firewall and current American DNS practice. In: Lecture Notes in Electrical Engineering. vol. 514, pp. 259–265. Springer Verlag (jun 2019). https://doi.org/10.1007/978-981-13-1056-0_27,