OFICIO No. 064/DGA/UNISTMO/2020
Asunto: *Gastos De publicación, Solicitud de Liberación.*
Sto. Domingo Tehuantepec, Oax., 19 de noviembre de 2020

Lic. Lorenzo Manuel Loera de la Rosa
Director de Superación Académica

AT´N:
Lic. Graciela Hernández Sánchez
Subdirectora de Análisis y Evaluación Docente

Por este conducto le envío un cordial saludo, al tiempo que remito la **Solicitud de Liberación** correspondiente al *apoyo para Gastos de Publicación* autorizado al **CA Cómputo Aplicado**, adscrito a la Universidad del Istmo. Cabe señalar que el apoyo fue autorizado mediante oficio No. 511-6/2020-1054, de fecha 07 de febrero de 2020.

Nombre del RCA	*Revista / ISSN*	*Título del artículo*	*Costo (M.N.)*
Dr. Daniel Pacheco Bautista	Fractals / 0218-348X	A NON LINEAR MODEL FOR A SMART SEMANTIC BROWSER BOT FOR A TEXT ATTRIBUTE RECOGNITION	$25,000.00

Se adjunta solicitud de liberación por parte del RCA del cuerpo académico en comento, Informe final (impacto académico logrado), copia del artículo publicado; y desglose financiero de recursos que emite la Universidad del Istmo. Los documentos en PDF se enviaron al correo victorh.osornio@nube.sep.gob.mx

Garantizando la transparencia en el ejercicio de los recursos, agradezco la atención prestada al presente, y aprovecho la ocasión para agradecer los apoyos que nos brinda el Programa en mejora de la educación de nuestra región, nuestro estado y por ende nuestro país.

ATENTAMENTE

Voluntas totum potest
Guiraa zanda ne guendaracala'dxi'

L.C.E. Claudia Hernández Cela
Jefa del Departamento de Gestión Académica

UNISTMO
UNIVERSIDAD DEL ISTMO
PROGRAMA PARA EL DESARROLLO PROFESIONAL DOCENTE, PARA EL TIPO SUPERIOR
PRODEP
REPRESENTANTE INSTITUCIONAL

C.f.p .-Dr. Modesto Seara Vázquez.- Rector de la Universidad del Istmo.- Para su conocimiento.
.-Dr. Israel Flores Sandoval.- Vice-rector Académico.- Universidad del Istmo.- Mismo fin.
.-Dr. Daniel Pacheco Bautista.- RCA Computo Aplicado y Profesor-Investigador.- Universidad del Istmo.- para seguimiento.
.- Archivo
*CHC/

Sto. Domingo Tehuantepec, Oaxaca. 05 de Noviembre de 2020

Asunto: gastos de publicación,
Solicitud de carta de liberación

M. en C. Guillermina Urbano Vidales
Directora de Superación Académica

AT´N: Lic. Graciela Hernández Sánchez
Subdirectora de Análisis y Evaluación Docente

Sirva el presente para enviarle un cordial saludo y agradecerle el apoyo recibido mediante el proyecto Gastos de Publicación. Así mismo, aprovecho la ocasión para solicitarle de la manera más respetuosa la **Carta de Liberación** correspondiente al recurso recibido en mérito del Programa, autorizado en el oficio No. 511-6/2020-1054, con fecha 7 de febrero del presente año. Me permito indicar que el artículo titulado *"A non-linear model for a smart semantic browser bot for a text attribute recognition"*, fue publicado en la revista *Fractals, Vol. 28, No. 2 (Marzo 2020),* con **factor de impacto JCR de 2.971** e ISSN 0218-348X.

Adjunto al presente, copia de dicho artículo, el cual se identifica mediante el DOI: https://doi.org/10.1142/S0218348X20500450.

Sin otro particular, agradezco su amable atención y quedo atento a su respuesta.

ATENTAMENTE.

"Voluntas totum potest"
Guirá' zanda ne guendaracala'dxi'

DR. DANIEL PACHECO BAUTISTA
REPRESENTANTE DEL CA CÓMPUTO APLICADO
UNIVERSIDAD DEL ISTMO

c.c.p. Dr. Israel Flores Sandoval.- Vice-rector Académico.- Universidad del Istmo.-Para su conocimiento.
c.c.p. Lic. Claudia Hernández Cela.- Representante Institucional PROMEP.-Mismo fin
c.c.p. Interesado.

Universidad del Istmo

INFORME FINAL

GASTOS DE PUBLICACIÓN

IES de adscripción:	Universidad del Istmo	**Título de la publicación:**	A non-linear model for a smart semantic browser bot for a text attribute recognition
Nombre del PTC:	Daniel Pacheco Bautista	**Nombre de la Revista:**	Fractals
ID del PTC:	UNISTMO-003	**ISSN:**	Print ISSN: 0218-348X Online ISSN: 1793-6543
Número de oficio de autorización:	No. 511-6/2020-1054	**Link: (opcional)**	https://www.worldscientific.com/doi/abs/10.1142/S0218348X20500450
Fecha de autorización:	7 de febrero de 2020	**Monto aprobado**	$25,000.00

Impacto académico logrado con el apoyo recibido:

El artículo propone un sistema disruptivo basado en la tecnología de cadenas de bloques (blockchain) utilizando un Bot experto para buscar y analizar información relevante que significativamente represente patrones de adopción de criptoconcurrencias. La revista en la cual fue publicado, es altamente especializado en el área de la computación con factor de impacto JCR de 2.971 y en la modalidad de acceso abierto, por lo que muchos investigadores alrededor del mundo podrán tener acceso al mismo, repercutiendo de forma importante en el desarrollo de múltiples aplicaciones basadas en la tecnología de cadenas de bloques. Como ejemplo, apenas a 7 meses de haberse publicado, la revista reporta más de 450 visitas al artículo y cerca de 112 descargas, procedentes de diversos países entre los que se encuentran: EEUU, Singapur, Corea del Sur y Japón.

Dr. Daniel Pacheco Bautista
Representante del CA Cómputo Aplicado

L.C.E. Claudia Hernández Cela
Representante Institucional ante PRODEP

UNIVERSIDAD DEL ISTMO

PROGRAMA PARA EL DESARROLLO PROFESIONAL DOCENTE, PARA EL TIPO SUPERIOR 2020

OFICIO APROBACION DSA 511-6/2020-1054 07'FEB'2020

GASTOS DE PUBLICACIÓN

prodep TIPO SUPERIOR

Beneficiario : DANIEL PACHECO BAUTISTA

Fecha	No. Ch. / T.E.	No. Comprobante	Beneficiario del Pago	R.F.C.	Concepto del Gasto	IMPORTE FACTURA (Ejercido)	GASTOS DE PUBLICACIÓN
			Monto Aprobado				$ 25,000.00
21/04/2020	80029	30736722	REVISTA " FRACTALS JOUMAL" ISSN 0218-348X	***	PAGO DEL ARTICULO " A NON-LINEAR MODEL FOR A SMART SEMANTIC BROWSER BOT FOR A TEXT ATTRIBUTE RECOGNITION"	$ 25,000.00	$ -
					Total Gastos	$ 25,000.00	$ -

Cd. Ixtepec, Oax. a 30 de Junio 2020

L.C. Yazmín Carolina López Morales
Jefa del Departamento de Recursos Financieros

UNIVERSIDAD DEL ISTMO UNISTMO OAXACA DEPARTAMENTO DE RECURSOS FINANCIEROS

L.C.E. Eugenio Cortés Hernández
Vice-Rector de Administración

UNIVERSIDAD DEL ISTMO UNISTMO OAXACA VICE-RECTORIA DE ADMINISTRACION

Fractals, Vol. 28, No. 2 (2020) 2050045 (11 pages)

DOI: 10.1142/S0218348X20500450

A NONLINEAR MODEL FOR A SMART SEMANTIC BROWSER BOT FOR A TEXT ATTRIBUTE RECOGNITION

R. CARREÑO AGUILERA,[*,§] M. A. ACEVEDO MOSQUEDA,[†]
M. E. ACEVEDO MOSQUEDA,[†] S. L. GOMEZ CORONEL,[†]
I. ALGREDO BADILLO,[‡] D. PACHECO BAUTISTA,[*]
M. PATIÑO ORTIZ,[†] J. PATIÑO ORTIZ[†]
and M. A. MARTINEZ CRUZ[†]

[*]*Universidad del Istmo, Cd. Universitaria S/N, 70760 Tehuantepec, Oaxaca, México*

[†]*Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatenco, México City, México*

[‡]*Instituto Nacional de Astrofisíca, Óptica y Electrónica INAOE, Puebla, México*

[§]*ricardo.carreno.a@hotmail.com*

Received May 21, 2019
Accepted December 17, 2019
Published February 14, 2020

Abstract

In spite of the advances in the state of the art in semantic artificial intelligence applications, there is still a long way to go to bring it to a level of mass adoption. Thus, in order to contribute to the advancement of this topic, this study develops a feasible model with a potential scalability for semantic applications' mass adoption, specifically for news or statement cluster attribute identification, either positive, negative or neutral. This paper proposes a disruptive system based on Blockchain using a Semantic Browser Expert System Bot with artificial intelligence called Blockchain Semantic Browser Expert System (BSBES) to look for and analyze relevant information that significantly represents the cryptocurrencies adoption patterns. The artificial intelligence in this study consists of a deep learning neural network to process the input information to identify the news pattern in a semantic way using deep learning based on two aspects of the news: technical aspect and adoption aspect of the cryptocurrencies. BSBES performance is achieved based on deep learning tools, and scalability is supported by a blockchain system including a stability study.

Keywords: Blockchain; Expert System; Hashtag; Semantic Browser.

1. INTRODUCTION

This research belongs to the computational sciences field, particularly to artificial intelligence (AI), because it refers to an algorithm based on a semantic computational AI browser for the determination of positive, neutral or negative patterns. These patterns are based on cryptocurrencies web news semantic browser, looking for two relevant aspects such as technological-innovation development and adoption (usage) for main cryptocurrencies. Speaking of technological-innovation developments, it is important to measure the usage of the number of queries done by users to this application, the objective is to measure the capacity and analyze the stability of this application to support a big scale of users in the future. Therefore, we have a Blockchain network connecting electronic devices for the worldwide scale. The semantic expert system uses AI with a bot (smart software)[1–3] by deep belief learning to do the semantic analysis. Different types of deep belief learning[4–7] are analyzed to choose the best to fit the model.

With the hype of the cryptocurrency market, the necessity for methods to detect positive, negative and neutral news patterns is increasing, especially due to high volatility. Even with the use of the existing browsers it is not possible to get sufficient information related to news searching in a quick and smart way. This is the reason a semantic (smart) search is important when searching big data information obtained from news and different web pages that satisfies our query is classified and tagged as positive, negative and neutral (indecision). Therefore, the objective of this research is to provide a Blockchain Semantic Browser Expert System (BSBES) to assist people interested in checking news trends on cryptocurrencies for a specific purpose or just to be updated about news. Furthermore, this BSBES can be a reference for every new development in the semantic field.

In this research, we use deep learning with a validated learning representation using probability and statistics for the first input layer. This layer is totally or partially used in other layers.

In this paper, we present the deep architectures as an alternative in the pattern classification and identification of dynamic systems problems. We also expose the historical problems that diverse learning algorithms have had when being applied to an elevated number of abstraction layers from the input dataset.

The overfitting[8–10] is one of the various challenges faced during the learning process of a neural network since it was observed an overlearning of a particular region in the parameters space dedicated only in one same region, that is why we can affirm that in this case the distribution or generalization of the unsupervised training of the first layer to the other ones failed. Another problem that the deep learning training faces is the local minimum problem. In other words, the cost function gets to a local minimum and the learning process stops due to the fact that the algorithm decides that the global minimum has been found to be an incorrect conclusion. To overcome this minimal apparent problem, auto-encoders (Boltzmann machines) methods[11–13] are used. Examples of this are: stochastic descending gradient training, noise addition in the code, shortage condition in the code and the verisimilitude maximization of the generative model to do a Boltzmann learning a random learning algorithm derived from ideas based on mechanical statistics, whose objective is the modeling of a probability distribution specified by the applied patterns in the visible neuron inputs (input neurons).

Based on the concept of the knowledge society,[14–16] we can find some elements that lead us to find a new communication architecture and, in conjunction with the concept of semantic browser,[17–19] and blockchain,[20–24] a multidisciplinary method is developed in this study. The algorithm of this method provides an impact value to a BSBES to this knowledge field, especially because it is based on an intelligent browser for proper news searching in social media networks with the assistance of AI.

This represents a real challenge. Given the fact that even an information search, made by a human expert is not efficient enough to determine the cryptocurrency news pattern as fast as he would like to do it, we propose an automatically search and collect information with the use of deep learning neural networks. Also, gathering information from the Internet in an intelligent way requires the inclusion of the concept of the semantic web, (still in the process of being developed) to provide satisfactory results to be used in a massive adoption. The contribution of this paper is in the semantic field using AI specifically on blockchain applications with cryptocurrencies attributes analysis.

2. BSBES EXPERT SYSTEM DESIGN DEPLOYMENT

The BSBES system has a reference keywords database for searching cryptocurrency news online. The search is done based on a web page list that the system uses to do the semantic search. A keywords hash is created for every single keyword found in the semantic search. A hash is associated to a set of keywords found with meaning (semantic) that matches a pattern database phrase. The chosen phrase is considered to be the optimal selection to fit the meaning and it is used for the final arrangement of phrases to determine the assessment of the news under semantical analysis. The assessment can be either positive, negative or neutral to each cryptocurrency news item analyzed by the system. Based on Figs. 1 and 2, we can explain the mechanism of this BSBES for cryptocurrency pattern.[25–29]

2.1. Blockchain Semantic Browser Expert System Bot

The expert system performs the semantic searches through a device that can be a mobile device or a personal computer, which will serve as an interface between the user and the machine (Fig. 1).

The modules of the Semantic Browser Expert System are represented in Fig. 1, which are the basic components that the Expert System shell has. It also has a database that the neural network uses to validate if the word found is a keyword and if it is positive, negative or neutral, using a deep learning neural network as its inference engine.

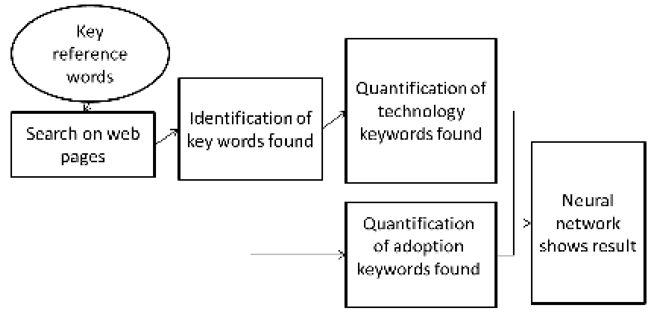

Fig. 2 Semantic browser expert system process.

Figure 2 shows the general mechanism of the BSBES Search Engine scheme. The function of the semantic search engine is to find technological aspects and features of adoption of cryptocurrencies dynamically, within a time interval, to know how these two aspects get worse or better in this interval. The semantic search engine will start with the first word of the text to initiate the process to match this word with a keyword, and the BSBES indicates to the semantic search engine when to continue this process with the next word. The inferential machine formed by the deep learning neural network will carry out the process of intelligent semantic association of the keywords found in the database. Two types of hashes (word and phrase) are generated to complete the whole semantic task as shown in Fig. 3. Every hash is a sequence of binary digits encoded with a block-chain algorithm (SHA256). A single-meaning hash (keyword) contains a single basic idea, and the sentence-meaning hash (keyphrase) is built by arranging a related group of single hashes. A group of related-sentence hashes build a paragraph, and a group of paragraphs complete the whole news analyzed online. This whole semantic hash process is performed for the two types of analysis: the technical and the adoption analysis. The technological aspects are

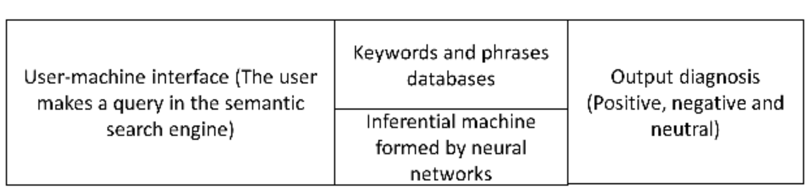

Fig. 1 Browser expert system modules.

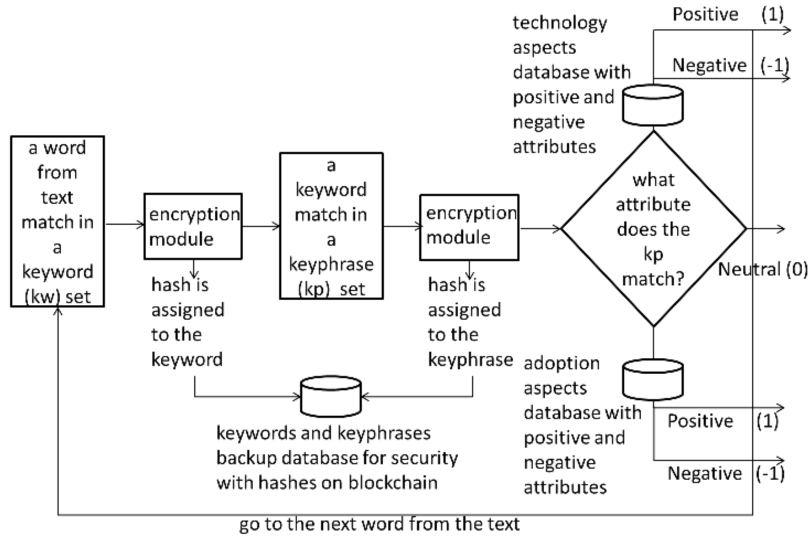

Fig. 3 Semantic keyword and keyphrases process.

quantified finding news related to software updates and technological characteristics of cryptocurrencies such as transaction per second (tps), transaction fees and scalability, such that the expert system fits the news into a pattern, using a semantic analysis. The technology and the adoption analysis aspect are determined by the technological and the adoption attribute quantity found in the news on open webs. If positive attribute is predominant, the text is marked as 1, if negative attribute is predominant, it is marked as -1, if no predominant is found, then it is marked with 0 (neutral). This same process for marking with 1, -1 and 0 is done for technological and adoption aspects.

The neural network obtains the inputs from both analysis (technological and adoption) to learn the outputs based on the logic combination shown in Fig. 4. This logical truth table determines the relationship that the inputs must have with the outputs when the neural network learns, then sends the results to the interface and the database. The BSBES displays the results of the search, providing valuable information regarding the market patterns of cryptocurrencies.

Figure 4 shows how the output neural network interact with the logical combination from the two aspects: technology and adoption. The neural network output can be a 1 (positive pattern), -1 (negative pattern) and 0 (neutral) result. Truth table for the inferential machine module (see Fig. 4) is shown in Table 1. In testing the expert system BSBES, 80% precision is obtained for adoption analysis and 60% precision for technical analysis.

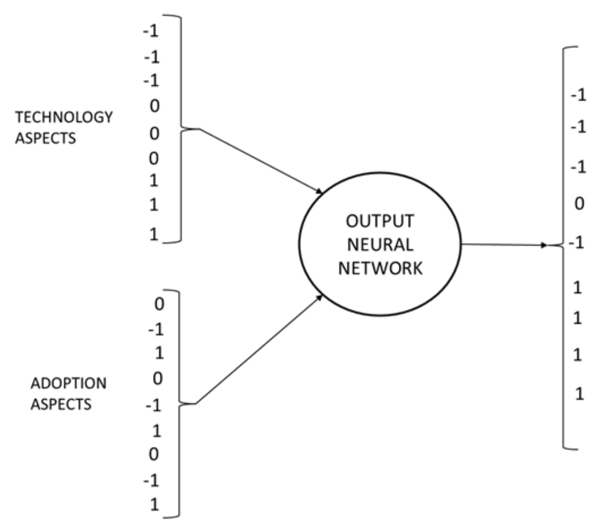

Fig. 4 Inferential machine module logical combination interaction.

Table 1 indicates the final attribute assignation: (1, -1, 0) considering the attribute (1, -1, 0) for each aspect, for example, if result for the text in adoption aspect is negative (-1) and for technological is neutral (0) the final result for the text is negative (-1), but the only result that matters is the final result because this is the result given to the user. Therefore, OUTPUT on Table 1 is the result given to the user.

Figure 5 shows the details of the "inferential machine" module, which includes two databases, and one deep neural network that helps the semantic search engine using a text comparator to match the text with keywords and then the blockchain

Table 1 Keywords Deep Learning Logic Truth Table.

Adoption Analysis Input 1	Technical Analysis Input 2	Output
−1	0	−1
−1	−1	−1
−1	1	−1
0	0	0
0	−1	−1
0	1	1
1	0	1
1	−1	1
1	1	1

will generate the hashes of the semantic elements: the single-meaning hash and the sentence-meaning hash. Both semantic elements are required by the inferential machine to process information semantically and smartly. Keyword and keyphrase database have thousands of possibilities to match to hundreds of words and phrases from the text under analysis, causing a lot of nonlinear relations between the inputs and outputs. Therefore, deep learning is needed to catch these nonlinear relationships.

Figure 5 also shows the interaction of the components of the BSBES with the required neural networks that structure the keywords in the form of a sentence, using the two hashes mentioned above and the logical combination (Fig. 4) to give the result. In this figure, it can be seen that the semantic neural network input are the keyphrases (in the form of sentence hash) chosen in the process to match keywords and keyphrases with the text using deep learning. Input and output of the output neural network is according to Fig. 4.

The innovative concept of "Blockchain" is driving the internet applications to the next level in the disruptive technologies. Based on the fact that not all blockchain applications have been successful in mass adoption yet, in this study the knowledge society is considered a new element to drive the current level of the blockchain to a new application as proposed in this paper. The contribution of this paper is in the semantic field using AI specifically on blockchain applications with cryptocurrencies attributes analysis; however, it may also help to the mass adoption of blockchain, using AI such that we can use some new semantic methods found in this research, as shown in Fig. 6, where we can

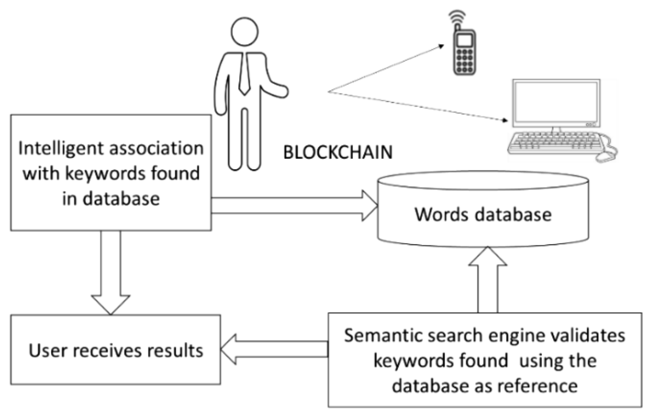

Fig. 6 Communication architecture of the Blockchain Expert System.

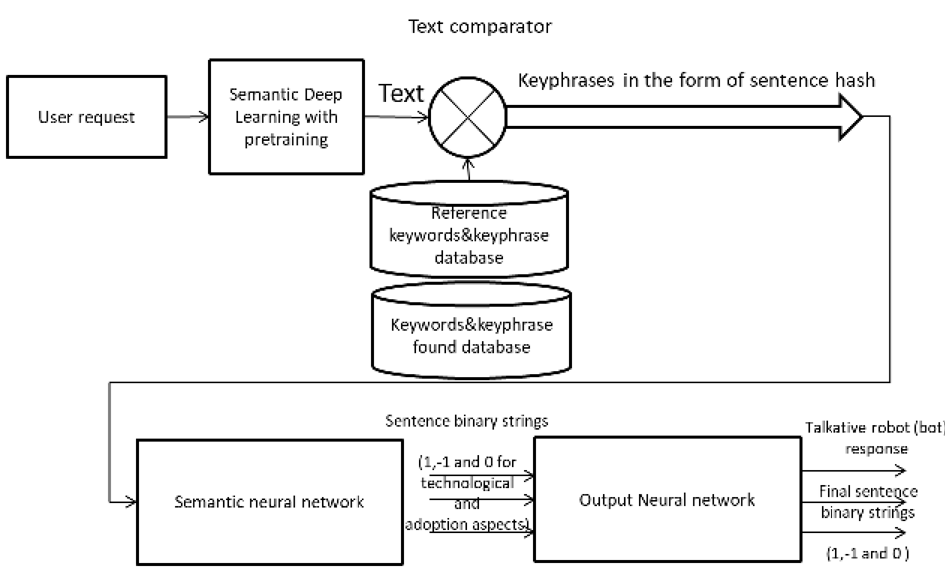

Fig. 5 Inferential machine flow chart of the Semantic expert system.

see how the user-machine interface module interacts with the intelligent semantic search engine system. Since a big amount of information (Big Data) is generated in the regular global application, in the same way, a big amount of information is generated related to technical and adoption aspects news.

2.2. BSBES Database

BSBES has two databases as shown in Fig. 3: keyword database and keyphrase database. Both databases are for reference; a word of the sentence under analysis must match to a keyword of the keyword database, and then the whole sentence under analysis must match with a keyphrase (sentence) of the keyphrase database; therefore, every keyword and every keyphrase is a semantic meaning. Since each keyphrase of reference is associated with a bipolar result (+, − or neutral), one of these three results will be given after the neural network associates the sentence under analysis with one of the keyphrases. Security is important in this development, therefore, a backup database is done assigning a hash to each keyword and each keyphrase (a hash is represented by an encrypted code in a blockchain). Since two types of attributes are wanted (technical and adoption), there is a dataset for each attribute ((+, − or neutral) in the keyword database and keyphrase database. The list of the page webs was chosen as the most visited in this field according to google. Database is not yet an open-access data to the public. The purpose of the keywords is to have a word that represent a set of similar words in order to simplify the whole process, and a hash is created and associated to each keyword in order to keep safely data on blockchain; therefore, after hashes for each keyword and keyphrase is created, it is practically the same to talk about the keyword hash and the keyword because they keep associated in the code permanently. The reason to create hashes is to make possible to use a blockchain environment.

2.3. Deep Learning Neural Network Development

In a multilayer-perceptron neural network, the k layer delivers an output vector h^{k-1} using the h^k output obtained from the last layer, beginning this process with the $x = h^0$ input. A classical activation example for a neural network function is the hyperbolic tangent whose output's layer is given by Eq. (1):

$$h^k = \tanh(b^k + w^k h^{k-1}), \tag{1}$$

where b^k is the offsets vector and w^k is the weights matrix. The activation function can be changed depending on the problem to be solved. Another very common function is the sigmoid, given by Eq. (2):

$$\text{sigm}(u) = \frac{1}{2}(\text{sigm}(u) + 1). \tag{2}$$

Nevertheless, for this study we use the tangential hyperbolic function for the hidden layers, due to the better results obtained in the learning; and a sigmoid function as the activation function for the output layer. The total output network h^l is expressed by the optimization function (cost function) $L(h^l, y)$, due to the fact it is typically convex for $b^l + w^l h^{l-1}$. Thus, it can be represented by Eq. (3):

$$h_i^l = \frac{\exp(b_i^l + w_i^l h_i^{l-1})}{\sum_j \exp(b_i^l + w_i^l h_i^{l-1})}, \tag{3}$$

where $\sum_i h_i^l = 1 y h_i^l$ is positive.

The h_i^l output can be used as an estimator[30,31] of $P(Y = y \mid x)$. In this case, a negative value of the verisimilitude logarithm is used; therefore, $L(h^l, y) = -\log P(Y = y \mid x) = -\log h_y^l$ in which the expected value for (x, y) has to be minimized with the cost function.[32–34] The sigm(u) transference function is used to design the neural network with extreme learning machine (ELM) architecture, where, by definition, the hidden layers and the output layer are trained differently. This ELM neural network is the architecture of the neural network used once a deep learning low dimension representation is found by Belief Learning statistics methods.

There are two stages during the training in this research; the first is a selfish layer pre-training and the second is supervised learning.

In the pre-training stage, iteration on all model layers is done using Boltzmann machines, then the parameters layers are guided to better regions into the parameter space. Once all the neurons are pre-trained, the neural network is adjusted using the descendent gradient method.

A function is implemented on each layer to execute a descendent gradient algorithm to optimize the weights to reduce the errors on each layer. This function is applied using the training dataset in a training-fixed epoch number. Pre-training algorithm with the DBN structure uses a Gibbs step with fine adjustment with descendent gradient an early stop with a 7% improvement.

A single training epoch is used in both stages as follows:

from $i = 1$*to* $i =$ *number of layers*:
while (*epoch* < *pretraining epochs*) : *epoch* = *epoch* + 1
on each minibatch in the training set:
pretrain layer [i](*input* = *current batch*) *retrieve parameters*

The purpose of the unsupervised pretraining is to drive the function to a restriction on the parameter space region where a solution is allowed, such that a regulatory behavior is granted.

Once all layers are pre-trained, the neural network goes to the second stage known as fine adjustment.

Three hidden layers architecture has 200 neurons on each, 19 pretraining epochs for each layer and 1500 unsupervised adjustment epochs with 0.1 and 0.01 learning rate, respectively.

Extreme Learning Machine model, ELM is used in the second stage.

2.4. Extreme Learning Machine Model

For N arbitrary distinct samples $(X_i; t_i)$, where $X_i = [X_{i1}; X_{i2}; \ldots; X_{in}]^T \in \mathbb{R}^n$ and $t_i = [t_{i1}; t_{i2}; \ldots; t_{im}] \in \mathbb{R}^m$, for ELM[35–38] with M hidden nodes and activation function $g(x)$, the Single Hidden Layer Feedforward Neural Networks (SLFN) output is mathematically modeled as

$$\sum_{i=1}^{M} \beta_i g_i(X_j) \sum_{i=1}^{M} \beta_i g_i(W_i \cdot X_j + b_i) = t_j, \quad \text{where: } j = 1, 2, \ldots, N, \tag{4}$$

where $W_i = [W_{i1}, W_{i2}, \ldots, W_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\beta i = [\beta i_1, \beta i_2, \ldots, \beta i_m]^T$ is the weight vector connecting the ith hidden node and output nodes, and bi is the threshold of the ith hidden node. Wi and Xj denote the inner product of wi and Xj. The results of the ELM are as shown in Fig. 7, where minimum error is reached quickly.

The parameters used in the nonlinear simplified network (ELM architecture) shown in Fig. 7 are:

- ELM two hidden layers quantity: 2
- ELM Inputs = 8
- ELM training data quantity = 999
- ELM testing data quantity = 10999
- ELM activation function of the hidden layers: tanh
- ELM hidden layer 1 neurons quantity: 10
- ELM hidden layer 2 neurons number: 10
- ELM output layer activation function: sigmoid
- ELM output layer neurons quantity: 19

2.5. BSBES Stability Study

Stability of the unvalidated hashtags must be ensured; therefore, a study on the dynamic stability of the total unvalidated hash tags ($X(t)$) of the stochastic process is performed. A valid hashtag sentence, means to have match semantically with the reference database, where $X(t)$ should be around a constant in order to avoid leaving unvalidated hashtags behind, and the $X(t)$ process should comply with the limit of the positive recurrence,[39,40] and the process of incoming hashtag validations should be modeled by a Poisson process with λ as the rate of this process and $\boldsymbol{h}$ as the average hashtag approval time. If we define expected cumulative

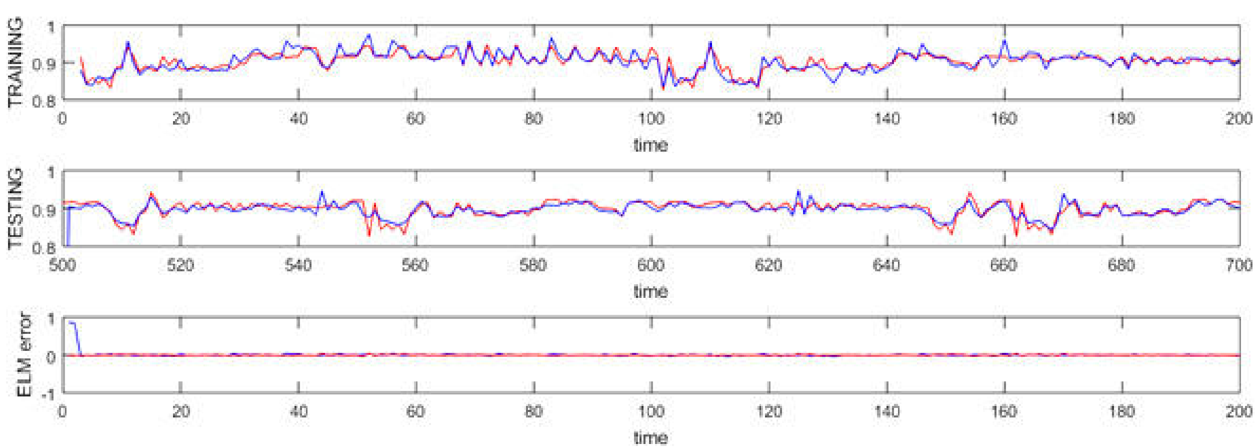

Fig. 7 Training and testing ELM training performance.

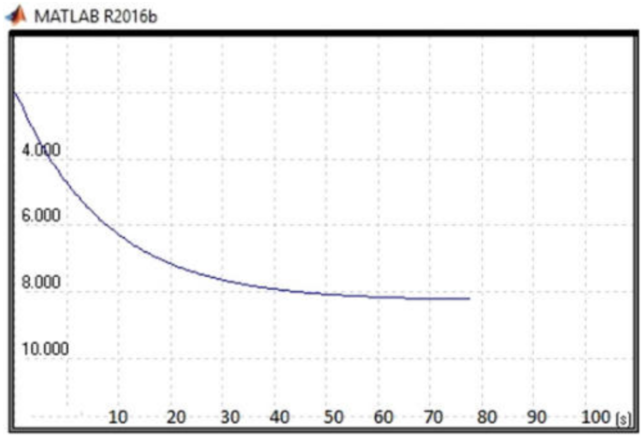

Fig. 8 Stochastic dynamic process stability for total unapproved hash tag sentences.

weight (ECW), then it can be represented by Eq. (5) as follows[41]:

$$\boldsymbol{d/dtECW(t) = ((\lambda k(t-h))/Xo) \times (2 - k(t-h)/Xo)} \ldots. \quad (5)$$

As per Eq. (5), we have the chart in Fig. 8.

In Fig. 8, we can see total unvalidated hash tag sentences going to a constant along time, which means that the total unvalidated hash tags tend to a limit such that there will never be unvalidated hash tag sentences without the proper validation for an indeterminate time.

3. RESULTS

3.1. Deep Learning on Blockchain

The development of the expert system has a one-terabyte database, a 7th generation Intel CORE I7 processor with a GTX 1070 GPU. A precision of 87.5% was obtained in the semantic search engine. The precision can be improved using a bigger database but this will reduce the learning speed; therefore, precision it is not only a neural network issue. However, the obtained precision is acceptable for this study if we consider that the core of this paper is the semantic neural network performance but not the database performance. Table 2 shows the comparative results using three common secure hash algorithms used for blockchain: SHA256 used in bitcoin protocol, SCRYPT used in Litecoin protocol and X11 used in dash coin.

Table 2 shows the performance in time of semantic searches where:

L represents the number of entries,

N represents the number of hidden nodes per layer,

M represents the size of the database and;

D the number of depth layers (deep learning dimension).

Parameters used in the deep learning net for this study are in Table 3:

The deep belief network helps to minimize the dimensionality effect problem in the deep architecture models. It is observed that by using only two identification layers we can have a minimal testing error; nevertheless, on the medium region of the surface, it is noticed that when increasing the number of layers and units, we can get to a new magnitude local minimal from the studied error (ideal deep dimension).

In Table 3, it can be seen the configuration with different neural network architectures and the error dynamic performance for the deep learning training is shown in Fig. 9, with a testing database. In this case, the training database and testing database are the same. In Fig. 9, it can be observed that minimum error can be obtained with nine layers; therefore, in this study for this application nine layers is recommended for deep learning.

Table 2 Blockchain Algorithms Comparative Performance.

Algorithm	Parameters	Training (seconds)	Testing (seconds)
(SHA256)	$L = 843$, $N = 10$, $M = 1$ terabyte, $D = 7$	35.8	29.33
(SCRYPT)	$L = 843$, $N = 10$, $M = 1$ terabyte, $D = 7$	27.2	24.5
(X11)	$L = 843$, $N = 10$, $M = 1$ terabyte, $D = 7$	42.5	39.1

Table 3 Hyperparameter of a Deep Network Range.

	Units Per Layer	Layers	Pre-Training Learning Rate	Fine Adjustment Learning Rate
Minimum	3	1	0.01	0.1
Maximum	10	7	0.1	0.3

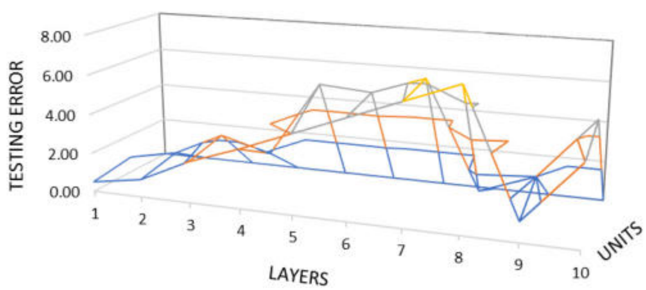

Fig. 9 Deep learning dynamic error.

Table 4 Comparative Table for ELM, MLP and DL.

	ELM	MLP	DL
COST ($\times 10^{-3}$)	11.012	19.175	7.965

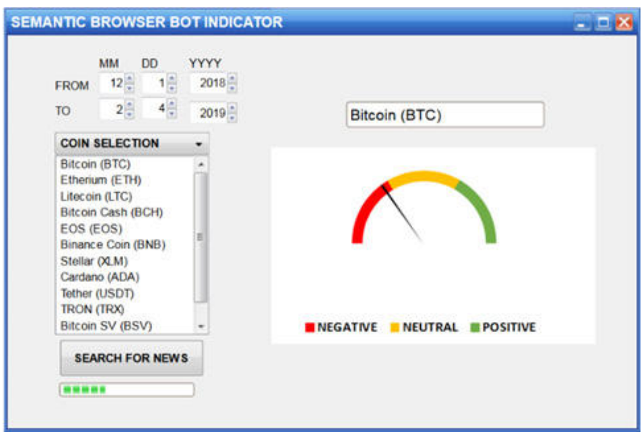

Fig. 10 Semantic browser indicator.

In Table 4, a comparative performance between three different neural network structures: ELM, Multilayer perceptron (MLP) and deep learning (DL) is shown.

3.2. BSBES User Interface

In Fig. 10, the Semantic Browser Bot indicator is shown. Note that period and cryptocurrency must be entered to ask the bot (software robot) for a semantic diagnosis, either: negative, neutral or positive. A click on SEARCH FOR NEWS button will run the expert system.

4. DISCUSSION

Expert systems have been successful in decision-making applications, for example in sickness diagnosis. However, semantic applications, in general, have a long way to go to achieve. Current semantic browsers just look for the words separately but they do not understand the whole statement by meaning (semantically). That is why this BSBES development proposes at least the interpretation of a whole statement in the positive, negative or neutral meaning. However, the biggest challenge to that the big data of sentences required to match all the possibilities of the meaning. The bigger and more accurate the sentences database, the better the results, but the training process becomes slower. That is why a neural network based on deep learning was proposed, and a blockchain platform has been implemented to keep data safe. Deep learning performance compared to ELM (extreme learning machine) and ML (multilayer) resulting in the best choice deep learning as shown in Fig. 9. Regarding blockchain protocol, several protocols were compared in performance as shown in Table 2, resulting in the protocol scrypt with the best performance in training. Also, a stability study was performed in Sec. 2.5 to ensure the stability of the model and in Fig. 9, we show the best dynamic error performance for nine layers.

5. CONCLUSIONS

A specialized semantic search engine, that is bounded to an information field, with a precision of 87.5% can be acceptable for an experimental and research purpose in order to prove the functionality of this study; this accuracy does not mean that the rest of the percentage is equivalent to incorrect semantic searches done, but it means answers close to the ideal answer. A close answer could be understandable or partially understandable in terms of semantic idea; therefore, this precision can be enough to understand the main idea of the semantic interpretation.

The minimum local problem was presented in this investigation. This was overcome by training every single layer from the network using an unsupervised pre-training algorithm in each one, starting with the network input layer (an autoencoder or Boltzmann machine was used for this purpose). So, the conclusion was that using this unsupervised pre-training algorithm in each layer is the best option.

An overfitting was observed in this study and to solve this problem, an unsupervised training input signal was placed into each layer, and to guide the layer patterns of the network to improve regions within the parameters space, we used an estimator.

The existing differences between the solution of the classification problem and the identification problem using the Boltzmann restricted machines as constructor block were identified. In the case of classification, it was realized that it is enough to consider binary visible inputs to the model. Meanwhile, this consideration delivers poor predictions when the system identification task with continuous inputs and outputs is considered.

As a future work, it is recommended to try an FPGA board (field-programmable gate array) to accelerate the central processing unit (CPU) process capacity to make semantic search and deep learning process faster since CPU speed is a constraint. Greater CPU capacity means larger databases can be used, thus improving the accuracy of results.

REFERENCES

1. D. A. White, Software engineering using artificial intelligence: The Knowledge-based software assistant, *Artif. Intell. Autom.* **1** (1998) 297–314.
2. Q. Wuang *et al.*, Research on application of artificial intelligence in computer network technology, *Int. J. Pattern Recognit. Artif. Intell.* **33**(5) (2019) 1–12.
3. W. Yu and X. Li, Automated nonlinear system modeling with multiple fuzzy neural networks and kernel smoothing, *Int. J. Neural Syst.* **20**(5) (2010) 429–435.
4. S. Kamada *et al.*, Knowledge extraction of adaptive structural learning of deep belief network for medical examination data, *Int. J. Semant. Comput.* **13**(1) (2019) 67–87.
5. H. Larochelle, Y. Bengio, J. Louradour and P. Lamblin, Exploring strategies for training deep neural networks, *J. Mach. Learn. Res.* **10** (2009) 1–40.
6. N. F. Hordri *et al.*, Deep learning and its applications: A review, *Postgraduate Annual Research on Informatics Seminar* (2016), pp. 1–5.
7. X. Hao *et al.*, Deep learning, *Int. J. Semant. Comput.* **10**(3) (2016) 417–439.
8. D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio and P. Vincent, The difficulty of training deep architectures and the effect of unsupervised pretraining, in *Proc. Twelfth Int. Conf. Artificial Intelligence and Statistics* (*AISTATS'09*) (2009), pp. 153–160.
9. X. Ying, An overview of overfitting and its solutions January, *J. Phys. Conf. Ser.* **1168**(2) (2019) 022022.
10. G. E. Hinton, S. Osindero and Y. Teh, A fast learning algorithm for deep belief nets, *Neural Comput.* **18** (2006) 1527–1554.
11. D. H. Ackley, G. E. Hinton and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cogn. Sci.* **9** (1985) 147–169.
12. G. E. Hinton and T. J. Sejnowski, Learning and relearning in Boltzmann machines, in *Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*, Vol. 1 (MIT Press, Cambridge, MA , 1986), pp. 282–317.
13. J. Baxter, A Bayesian/information theoretic model of learning via multiple task sampling, *Mach. Learn.* **28** (1997) 7–40.
14. A. Dattakumar and R. S. Sharma, Smart cities and knowledge societies: Correlation, causation or distinct?, *2016 IEEE Int. Conf. Management of Innovation and Technology* (*ICMIT*) (2016), pp. 193–197.
15. P. Weingart, The moment of truth for science: The consequences of the 'Knowledge Society' for society and science, *EMBO Reports* (2002), pp. 703–706.
16. S. Tobón *et al.*, Sociedad del Conocimiento: Estudio documental desde una perspectiva humanista y compleja, *Paradigma* **36**(2) (2015) 7–36.
17. S. Shirgave *et al.*, Semantically enriched variable length Markov chain model for analysis of user web navigation, *Int. J. Inf. Technol. Decision Making* **13**(4) (2014) 721–753.
18. S. X. Fang *et al.*, Web content visualizer: A visualization system for search engines in semantic web, *Int. J. Inf. Technol. Decision Making* **10**(5) (2011) 913–931.
19. M. Kopel, Libraries in the semantic web era, in *Qualitative and Quantitative Methods in Libraries* (World Scientific, 2010), pp. 115–122.
20. J. J. Xu, Are blockchain immune to all malicious attacks? *Finan. Innov.* **2**(25) (2016) 1–9.
21. M. Samaniego and R. Deters, Internet of Smart Things — IoST: Using blockchain and CLIPS to make things autonomous, *IEEE Int. Conf. Cognitive Computing* (*ICCC*) (2017), pp. 9–16.
22. E. Morgen, Blockchain world — Do you need a blockchain? This chart will tell you if the technology can solve your problem, *IEEE Spect.* **54**(10) (2017) 38–60.
23. T. Aste, P. Tasca and T. D. Matteo, Blockchain technologies: The foreseeable impact on society and industry, *Computer* **50**(9) (2017) 18–28.
24. J. Sun *et al.*, Blockchain based sharing services: What blockchain technology can contribute to smart cities, *Finan. Innov.* **2**(26) (2016) 1–9.
25. W. Gu *et al.*, Expert system for ice hockey game prediction: Data mining with human judgment, *Int. J. Inf. Technol. Decision Making* **15**(4) (2016) 763–789.
26. C. A. Magni *et al.,* An alternative approach to firms' evaluation: Expert systems and fuzzy logic, *Int. J. Inf. Technol. Decision Making* **5**(1) (2016) 195–225.

27. N. J. Pizzi *et al.*, Expert system approach to assessments of bleeding predispositions in tonsillectomy/adenoidectomy patients, *Adv. Artif. Intell.* **27** (1990) 67–83.
28. F. K. Flores, Internet of things: Managing wireless sensor network with rest API for smart homes, in *Theory and Practice of Computation* (World Scientific, 2016), pp. 132–142.
29. R. Carreño *et al.*, An IoT expert system shell in block-chain technology with ELM as inference engine, *Int. J. Inf. Technol. Decision Making* **18**(1) (2019) 87–104.
30. Y. Mao *et al.*, Velocity control of serial elastic actuator based on EKF estimator and neural network, in *Nature-Inspired Mobile Robotics* (World Scientific, 2013), pp. 729–739.
31. R. Carreño *et al.*, A nonlinear acoustic transducer model with variable structure control, *Fractals* **25**(2) (2017) 1–6.
32. S. Jafari, A new cost function for parameter estimation of chaotic systems using return maps as fingerprints, *Int. J. Bifurc. Chaos* **24**(10) (2014) 1450134.
33. W. Yu *et al.*, Neural networks for the optimization of crude oil blending, *Int. J. Neural Syst.* **15**(5) (2005) 377–389.
34. K. S. Narendra and K. Parthasarathy, Gradient methods for optimization of dynamical systems containing neural networks, *IEEE Trans. Neural Netw.* **2** (1991) 252–262.
35. W. Yu *et al.*, Modeling of gasoline blending via discrete time neural networks, *Neural Netw. Proc.* **2004** (2004) 6.
36. Y. Li, Extreme learning machine for determining signed efficiency measure from data, *Int. J. Uncertain. Fuzz. Knowl. Based Syst.* **21**(2) (2013) 131–142.
37. R. Carreño *et al.*, Comparative analysis on nonlinear models for RON gasoline blending using neural networks, *Fractals* **25**(6) (2017) 1–8.
38. H. T. Huynh *et al.*, An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks, *Int. J. Neural Syst.* **18**(5) (2008) 433–441.
39. Z. Rosberg, A positive recurrence criterion associated with multidimensional queueing processes, *J. Appl. Probab.* **17**(3) (1980) 790–801.
40. T. Crabill, Sufficient conditions for positive recurrence and recurrence of specially structured Markov chains, *Oper. Res.* **16**(4) (1968) 858–867.
41. S. M. Ross, *Introduction to Probability Models*, 10th edn. (Elsevier, 2010).