



Article

Deep Learning for Enterprise Systems Implementation Lifecycle Challenges: Research Directions

Hossam El-Din Hassanien *  and Ahmed Elragal 

Department of Computer Science, Electrical and Space Engineering / Faculty of Information Systems,
Luleå University of Technology, SE-971 87 Luleå, Sweden; ahmed.elragal@ltu.se

* Correspondence: hossam.hassanien@ltu.se

Abstract: Transforming the state-of-the-art definition and anatomy of enterprise systems (ESs) seems to some academics and practitioners as an unavoidable destiny. Value depletion lead by early retirement and/or replacement of ESs solutions has been a constant throughout the past decade. That did drive an enormous amount of research that works on addressing the problems leading to the resource drain. The resource waste had persisted throughout the ESs implementation lifecycle phases and dimensions especially post-live phases; leading to depleting the value of the social and technical dimensions of the lifecycle. Parallel to this research stream, the momentum gained by deep learning (DL) algorithms and platforms has been exponentially growing to fuel the advancements toward artificial intelligence and automated augmentation. Correspondingly, this paper is set out to present five key research directions through which DL would take part as a contributor towards the transformation of the ESs state-of-the-art. The paper reviews the ESs implementation lifecycle challenges and the intersection with DL research conducted on ESs by analyzing and synthesizing key basket journals (list of the Association of Information Systems). The paper also presents results from several experiments showcasing the effectiveness of DL in adding a level of augmentation to ESs by analyzing a large set of data extracted from the Atlassian Jira Software Issue Tracking System across different ecosystems. The paper then concludes by presenting the research directions and discussing socio-technical research courses that work on key frontiers identified within this scholarly work.

Keywords: enterprise systems; deep learning; implementation lifecycle; challenges; support tickets



Citation: Hassanien, H.E.; Elragal, A. Deep Learning for Enterprise Systems Implementation Lifecycle Challenges: Research Directions. *Informatics* **2021**, *8*, 11. <https://doi.org/10.3390/informatics8010011>

Academic Editor: Gian Antonio Susto

Received: 11 December 2020

Accepted: 11 February 2021

Published: 20 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rapid developments within the computing industry have resulted in a noteworthy dependency on computing technologies, for a variety of services. The existence of artificially intelligent technologies had significantly catalyzed the technological reach by applying the underpinning algorithms to serve a plethora of relevant use cases. Since 2006, the foundational concepts of deep structured learning, or more commonly known as deep learning or hierarchical learning, has emerged as a modern area of machine learning research [1]. During the past years, the witnessed advancements of deep learning methods had substantially influenced the aptitude of extracting complex patterns through several natural language processing (NLP) techniques. Given the propensity toward applying these techniques on big data, the potential for applying fast information retrieval, entity extraction, categorization, automatic data tagging, and simplifying selective tasks was made possible [2].

On the other hand, research conducted on enterprise systems (ESs), also known as enterprise resource planning systems or enterprise resource planning (ERP) systems, had seen significant advancements. These systems aim to produce integrated, modular, off-the-shelf systems aiming to control key functional areas within the enterprise such as sales, accounting and finance, material management, inventory control, and human resources [3].

The proposed benefits of using ERP systems are several—seamless information flow, access to real-time data, process-orientation, and improved communication across the enterprise.

Reaping the value out of an ES entails the pretext of going through a number of implementations that had been empirically described within two frameworks [4,5]. The one described by Esteves and Pastor ends constitutes six phases that are made up of adoption, acquisition, implementation use and maintenance, evolution, and end with retirement. The close within the retirement phase postures a number of good reasons describing the lifecycle for which some of them are caused by a set of problems leading to early retirement and/or replacement of the solution across the different implementation phases. Many of these are experienced through lack of training and inability to understand the main design principles of an ES, which typically leads to workarounds and customizations [6,7].

However, ERP implementation challenges have proved to have high organization and technical complexity, and the human consequences and required changes in business processes are often underestimated [8]. Correspondingly, scholarly evidence exposes various pretexts countering the propensity of realizing ERP value and incurring hidden costs across the ERP implementation lifecycles [9,10]. It is to be noted that this debate applies to both large enterprises (LEs) and small-and-medium-sized enterprises (SMEs), leading to what is known as the “*productivity decline*” or “*lack of continuous*” [11,12]. This then poses directions toward a number of dimensions to be discussed revolving around the dilemma of avoiding ERP misfits [13]. Or entertaining the typical exercise of bending the firm’s core business procedures and process to fit ERP software [9].

That being said, this paper aims at investigating the top determinants behind the ESs challenges across the ESs implementation lifecycle phases and the dimensions at which DL technologies would address, whereby the promised efficiencies that ESs instigate are to be realized.

This paper is structured as follows: Section 2 introduces the research background from the perspective of both ESs and DL technologies. This section sheds light on the anatomy of deep learning technologies in the first subsection and the ESs implementation lifecycle with scholarly definitions of the phases. Section 3 works on discussing the adopted research methodology formulating the guideline for the literature review process around ESs implementation lifecycle. This is also where the research questions guiding the directions of the literature review process are introduced. Section 4 takes on the exercise of analyzing the reviewed literature to produce the deduced observations formulating the arguments out of the literature review exercise. Section 5 showcases a small-scale experimental example of how deep learning methods would be of benefit for ESs augmentation. Section 6 introduces the five top research directions laying out the foundations for future socio-technical research advancements set to avoid the phenomenon of ESs productivity decline.

2. Research Background

The following two sections are going to shed light on principle academic foundations that should work on sewing the hypothetical fabric of this literature review paper. These sections will therefore set the stage for discussions around the premises to be investigated. Hence, the discussions are weaved again altogether as part of the analysis and synthesis sections versus research questions that are to be investigated, tying all of them together into the final research agenda. For this purpose, we discuss the foundational building blocks of the deep learning architectures and the enterprise systems implementation lifecycle. These discussions should pave the way toward applying a systematic literature review process to uncover the research agenda questions pertaining to how deep learning architecture would be of use to addressing ERP misfits across the entire ERP implementation lifecycle.

2.1. Deep Learning: A Brief Technology Outlook for Text Classification and Information Retrieval

The definition of deep learning has been a challenge for many, given the slowly changing development of the technique over the past decade. One useful definition that explains the technique as one that is “about learning multiple levels of representation and

abstraction that help to make sense of data such as images, sound, and text" [1]. The main traits deep learning methods are characterized by are having more neurons than traditional neural networks, having rather more complex architectures of connecting layers/neurons in the underlying neural networks, requiring an explosive amount of computing power to be available to train, and having the ability of automatic feature extraction. The four major deep learning architectures are unsupervised pre-trained networks (UPNs), convolutional neural networks (CNNs), recurrent neural networks, and recursive neural networks [14]. The hierarchical nature of deep learning techniques facilitates the process of learning feature hierarchies from higher levels of the hierarchy formed by the conformation of lower-level features [15]. The compound and convoluted nature of the technique takes place at multiple abstraction levels allowing the system to learn difficult and dense patterns. Automatically learning features at several levels of abstraction takes place without relying on human-influenced features [16]. This is typically vital given the tedious human task of explicitly specifying high-level abstractions from incoming raw inputs [17]. This is especially important for higher-level abstractions in which humans often face significant issues specifying explicitly in terms of raw sensory input given the complex nature. The ability to automatically learn powerful features will become increasingly important as the amount of data and range of applications to machine learning methods continues to grow.

The emergence of deep learning architectures was triggered by the fact that traditional machine-learning methods were limited within their abilities to process natural raw data [18]. For long periods of time assembling a pattern-recognition or machine-learning solution required sophisticated engineering and considerable domain expertise to develop a feature extractor that transforms raw data into a suitable internal representation (feature vector) from which the learning architecture would detect or classify patterns from the input [19].

Correspondingly, deep learning architectures had introduced significant advancements to text mining and natural language processing [20]. The build of deep networks extends basic feed-forward multilayer neural networks in many ways. Typically, deep networks join smaller networks as building blocks into larger networks; in other cases, they use a specialized set of layers [14]. The usual building blocks are feedforward multilayer neural networks, autoencoders, and restricted Boltzmann machines (RBMs). Feedforward neural networks are the most prominent and simplest-to-understand neural networks. They have an input layer, one or many hidden layers, and a single output layer. Every layer can have multiple numbers of neurons, and each layer is fully connected to the adjacent layer [21]. On the other hand, autoencoders are simple learning circuits for which their objective would be to transform inputs into outputs with the least possible amount of distortion [22]. While conceptually simple, they play an important role in machine learning. "Autoencoders" have taken center stage again in the deep architecture approach where autoencoders become particularly useful to form "restricted Boltzmann machines" (RBMs) [22–25]. Correspondingly, RBMs are a category of neural networks that are being used to learn features from datasets in an unsupervised manner. This takes place by mapping input data to a hidden state and then aim to reconstruct the input from the hidden state [26,27]. On the other hand, deep belief networks (DBNs) are made up of layers of RBMs for the pre-training stage and then a feedforward network for the fine-tuning stage (Figure 1).

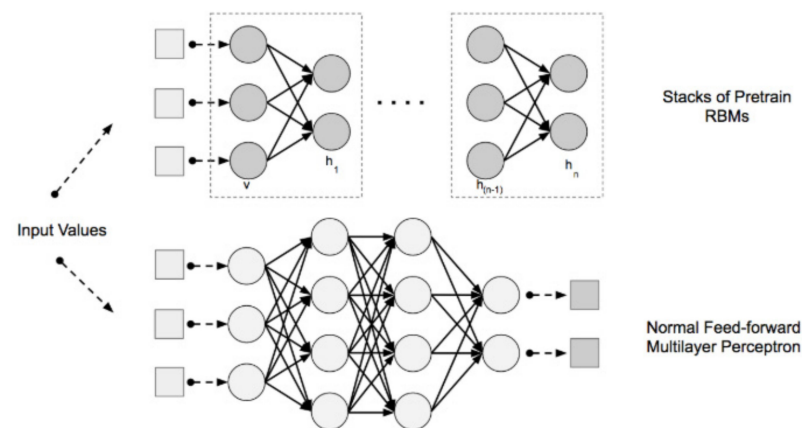


Figure 1. Deep belief networks (DBNs) architecture [14].

Applications of deep belief nets (DBN) addressing several problems and use cases triggered many research discussions from image classification and speech recognition to audio classification. With an example of applying DBNs to address a natural language processing (NLP) use case, DBNs did show a rather favorable classification performance when compared to other widely used learning techniques, like maximum entropy and boosting-based classifiers [28]. The architectural formations of DBN networks played a noteworthy role in the rise of many offspring deep learning architectures. These architectures led to significant breakthroughs addressing a plethora of use cases across the technology board. Likely, the efficiencies introduced by hybrid BDN models had also supported the accuracy enhancements for text classification applications [29].

When it comes to the higher-order understanding of discrete text features, CNNs did show how these models could learn from a dictionary of word inputs (words, phrases, sentences, or any other syntactic or semantic structures) related to a particular language without artificially embedding knowledge about these corresponding inputs [30]. The success of this approach for this particular use case (text understanding from scratch) was inspired by the earlier success of CNNs with various image recognition applications that learned from hierarchical raw pixel representations [31]. However, in this case, the words, phrases, and sentences replaced pixels in order to deliver a model that understands the text. It is, however, worth mentioning that in the case of CNNs, shallow-and-wide networks at the word level is still considered most effective for text classification and sentiment analysis. Deep CNN architectures, on the other hand, seldomly top shallow networks when text is encoded as a sequence of characters for obvious reasons [32].

Lastly, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) did also see many applications empowering text mining and NLP-based use cases. RNNs are types of feedforward neural networks. They stand out from other feedforward networks in their ability to send information over time-steps. The following is a brief explanation of RNNs from Juergen Schmidhuber [33]:

RNNs allow for both parallel and sequential computation, and in principle can compute anything a traditional computer can compute. Unlike traditional computers, however, Recurrent Neural Networks are similar to the human brain, which is a large feedback network of connected neurons that somehow can learn to translate a lifelong sensory input stream into a sequence of useful motor outputs.

In the world of information retrieval (IR) and text classification, several fundamental problems are being addressed, i.e., matching, translation, classification, and structured prediction [34]. It is quite evident that deep learning models can learn better representations for matching and other problems when compared to traditional text mining methods, which makes deep learning methods particularly effective for hard IR problems [35]. Among the main deep learning tools of IR and text classification are word embeddings, RNNs, and CNNs [36].

A core component of the neural network models handling NLP-based problems is the use of embeddings where features are represented as a vector in a low dimensional space. A word embedding is a learned representation for text in which words that have the same meaning have a similar representation [37]. One of the most prominent and widely adopted algorithms that transform words/letters into numbers is “word2vec” [38]. The algorithm is a two-layer neural net that processes text by vectorizing words. Its input is a text corpus and its output is a set of vectors that represent words in that corpus. The distinguishing aspect of the algorithm is that it groups the vectors of similar words together in vector space by detecting mathematical similarities, hence enabling the algorithm to detect relevant word contexts [39].

Siamese Manhattan LSTMs (MaLSTMs), on the other hand, are a rather special breed of the LSTM architecture that had witnessed fairly stable results in arriving at the semantic similarity between sentences [40]. In this deep neural network architecture, there are two networks—LSTM(a) and LSTM(b). Both LSTM(a) and LSTM(b) process one of the sentences in a given pair. Each sentence is passed to the LSTM and updates its hidden state at each sequence index. The final representation of the sentence is encoded by the last hidden state of the model. For a given pair of sentences, the model applies a pre-defined similarity function to their LSTM representations. Similarities in the representation space are subsequently used to infer the sentences’ underlying semantic similarity (Figure 2).

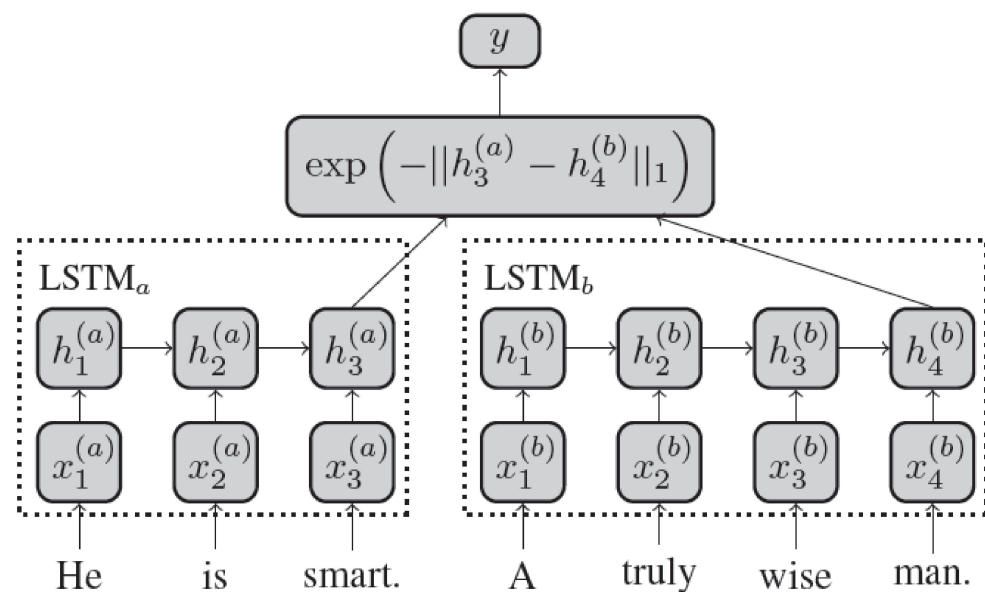


Figure 2. Deep belief networks architecture [14].

This model uses pre-trained input word-vectors on an external corpus in the form of word embeddings generated by the “word2vec” method that captures rather complex inter-word relatedness across the word vector space [38]. This architecture demonstrates the capabilities of modeling complex semantic relatedness.

2.2. Deep Learning Platforms

A survey of deep learning platforms, applications, and research trends published in the year 2018 had elucidated a number of significant facts around the currently existing deep learning technologies [41]. That had been followed by a few survey articles to probe DL architectures for big data [42] and DL architecture in the light of emerging cloud architecture [43] and the difficulties of deployment caused by compilers [44].

In brief, the discussion throughout these surveys stated some facts about the DL state-of-the-art being pursuing the trajectory toward plateaued maturity with fluctuating leadership that gets to be shared among many of the surveyed platform [41]. On the other hand, it was very clear the DL for big data is still in its infancy even though the

architecture provided by the DL architecture is possibly the best there is to support accurate representations compared to traditional advanced analytics algorithms [42]. Again, the infancy of the DL state-of-the-art had been constrained to the big data domain and extended toward cloud architectures [43], all of which either directly or indirectly point to the open challenges to accurately rationalize the effectiveness of optimizations adopted by different DL compilers [44]. All in all, this applies to platforms such as TensorFlow, Keras, DeepLearning4J, Apache MXnet, Microsoft Cognitive Services, Theano, Torch, and Caffe.

The fact that all of these platforms have been maintained by tech giants while primarily relying on the open-source communities meant that surveys conducted by these community sources could not have been ignored. Sources such as KDnuggets and Wikipedia need to be monitored to delve into the facts about these solutions [42,43]. By looking at both sources, it could be found that there are other significant platforms that do exist with highly varying adoption thresholds that consistently changes over time, which is quite evident in the case of PyTorch specifically with a delta adoption change with almost 76% increase, as seen in Table 1 and Figures 3 and 4. The most recent KDnuggets report published at the time our scholarly work had been developed studies all the above platforms in addition to others in correspondence to the adoption trends between 2018 and 2019, which illustrates the delta adoption change

Table 1. Deep learning platforms adoption survey, as conducted by KDnuggets [45,46], with platforms addressed by [41] highlighted in orange.

	Platform	2019 % Adoption	2018 % Adoption	% Adoption Change
1	Tensorflow	31.70%	29.90%	5.80%
2	Keras	26.60%	22.20%	19.70%
3	PyTorch	11.30%	6.40%	75.50%
4	DeepLearning4J	2.50%	3.40%	−25.60%
5	Apache MXnet	1.70%	1.50%	13.10%
6	Microsoft Cognitive Toolkit	1.60%	3.00%	−45.50%
7	Theano	1.60%	4.90%	−67.40%
8	Torch	0.90%	1.00%	−6.10%
9	TFLearn	0.70%	1.10%	−34.70%
10	Caffe	0.60%	1.50%	−58.30%

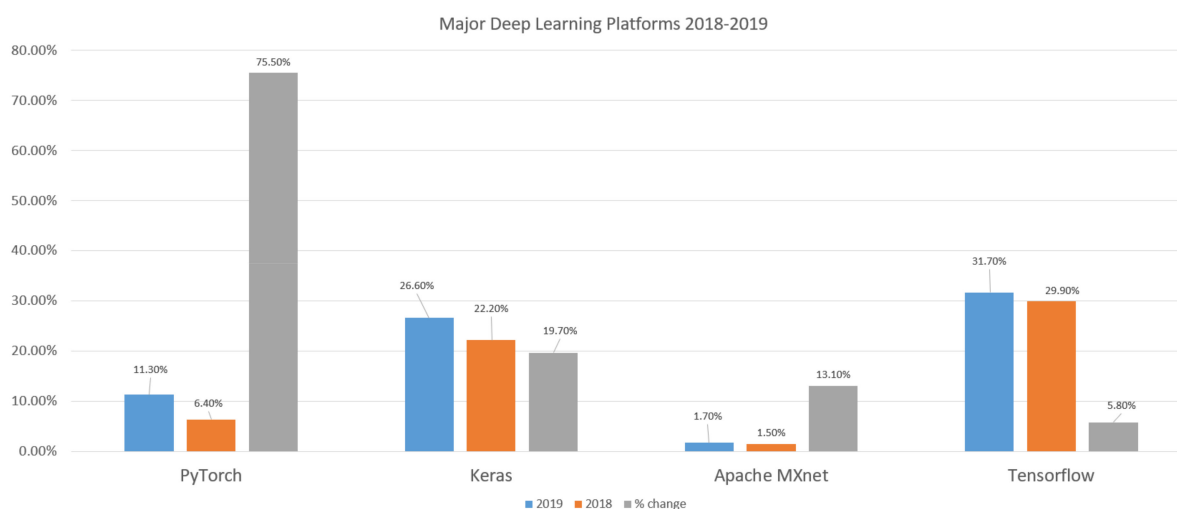


Figure 3. Graphical representation of the highest adoption change (sorted by adoption change %) of deep learning platform as conducted by KDnuggets poll 2018–2019 [39].

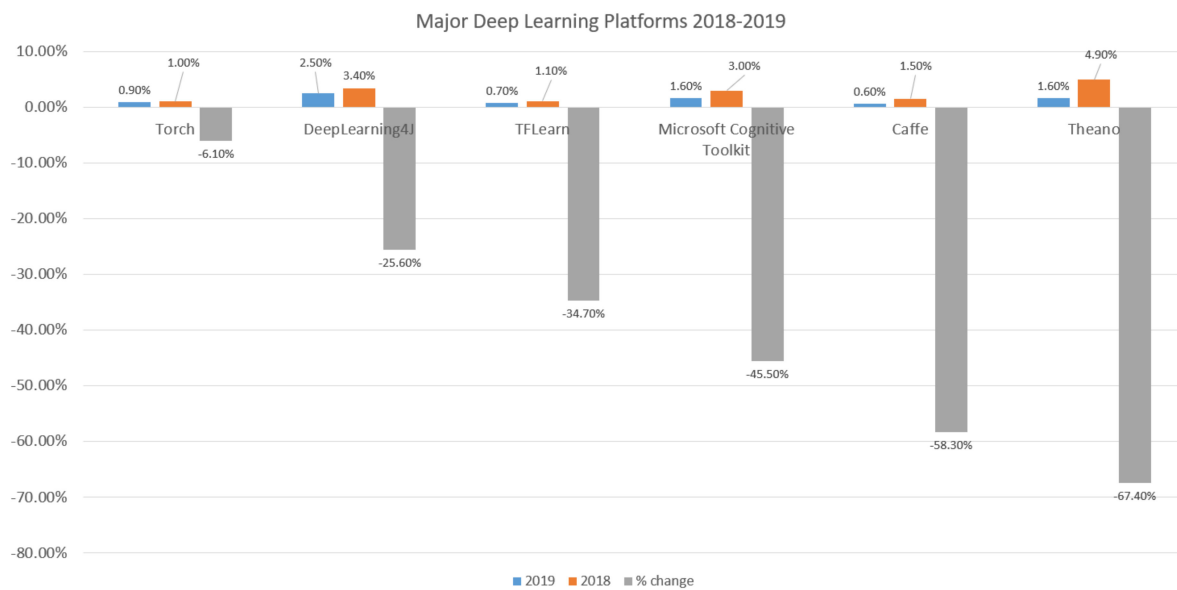


Figure 4. Graphical representation of the lowest adoption change (sorted by adoption change %) of deep learning platform as conducted by KDnuggets poll 2018–2019 [39].

Even though all of the above platforms provide comprehensive capabilities to run deep learning architectures, reusability and adaptability to the code pose significant level complexities. Finding the right model architecture and hyperparameters to train a DL model is a difficult aspect of any deep learning pipeline. Both data science practitioners and researchers spend hours experimenting with different settings and architectures to find the perfect fit DL model for their corresponding problem. Henceforth, this brings us to a new breed of DL platforms that we call DL representational frameworks (DL-ReFrams).

DL-ReFrams are a breed of DL platforms that adopt zero code methods to minimize the time consumed to develop and reuse state-of-the-art DL architectures through representational coding and Graphical User Interface (GUI) based techniques into prototyping and model deployment. These are new breeds of DL frameworks that address the shortage of data science and specifically deep learning code skills. At this moment, these frameworks are ones that are still on the rise of research and development. There are a number of pursued efforts that are working toward the realization of a comprehensive DL-ReFram. Looking at the research arena, MatchZoo serves as a DL-ReFram example that “facilitates designing, comparing and sharing of deep text matching models” [47]. On the other hand, if we look at the data science industry, we can also find the working example of Uber Ludwig [48]. Uber Ludwig is an open-source deep learning framework supported by Uber AI labs. It is built on top of TensorFlow that allows users to create and train models without writing code. DL-ReFrams typically share some of the following traits:

- No coding required: no coding skills are required to train a model and use it for obtaining predictions;
- Generality: a new data type-based approach to deep learning model design that makes the tool usable across many different use cases;
- Flexibility: experienced users have extensive control over model building and training, while newcomers will find it easy to use;
- Extensibility: easy to add new model architecture and new feature data types;
- Understandability: deep learning model internals are often considered black boxes, but we provide standard visualizations to understand their performance and compare their predictions.

Despite the useful traits of these platforms, it could still be observed that development is still in its early stages. Correspondingly, this does not position these platforms for mainstream adoption in either practice or research given the limited set of supported

DL architectures, hence making these platforms useful only if the supported DL models/architectures are provided by the corresponding platform.

2.3. Enterprise Systems Implementation Lifecycle

ESs lifecycle phases differ in names, number, and level of elaboration from model to model. However, most of these models usually include a number of phases that describe the pre-implementation, implementation, and post-implementation stages [49]. Many scholarly works within the field have exhausted the analysis of the ERP system development lifecycle whereby several prominent models appear. Research conducted by Esteves and Pastor [5] and Markus and Tanis [4] mark some of the important lifecycle frameworks. The following snapshot illustrates the proposed framework introduced by Esteves and Pastor [5], where the phases and dimensions are shown (Figure 5).

Within this framework, the phases describe the different stages a given ERP system lifecycle undergoes in typical organizations (adoption decision, acquisition, implementation, use and maintenance, evolution and retirement) [50]. On the other hand, dimensions describe the different viewpoints by which a given phase could be analyzed (product, process, people, and change management).

Scholar publications explain ERPs as a part of enterprise systems (ESs) [4]. ESs are commercial software packages that enable the integration of transitions-oriented data and business processes throughout an organization. ESs include ERP software and such related packages as advanced planning and scheduling, sales force automation, customer relationship management, and product configuration. The authors claim that the key questions about ESs from the perspective of an adopting organization's executive leadership are questions about success. Additionally, it has been claimed that no one measure of ESs success is sufficient for all the concerns an organization's executives might have about the ESs experience. Consequently, a "balanced scorecard" of success metrics addressing different dimensions at different points in time needs to be adopted governed by the ESs experience cycle phases. Organizations' experience with an ES can be described as moving through several phases, characterized by key players, typical activities, characteristic problems, appropriate performance metrics, and a range of possible outcomes. The phases within this cycle are explained as follows: the "chartering phase" comprises decision leading up to funding of an enterprise system; the "project phase" comprises activities intended to get the system up and running in one or more business units; the "shakedown phase" is the organization's coming to the adoption state of the implemented ES. Activities include bug fixing and rework, system performance tuning, retraining, and staffing up to handle temporary inefficiencies; and the "onward and upward phase" continues from normal operation until the system is replaced with an upgrade or a different system. During this phase, the organization is finally able to realize the benefits of its investment. Activities of this phase include continuous business improvement, additional user skill-building, and post-implementation benefit assessment; however, these typical activities are often not performed. With a focus on Esteves and Pastor [5] model as depicted in Figure 5, the implementation lifecycle stages are explained as follows:

1. Adoption decision: During this phase, companies begin to investigate for an ERP system that suits their corresponding business challenges to uplift the organizational performance. This phase then defines the shortlisting of ESs that fit the system requirements, goals, and benefits and a study of the impact of adoption at a business and organizational level [51];
2. Acquisition: Selection of the ESs product that is most appropriate for the organization takes place during this phase. With an aim toward diminishing the need for customization, a vendor and/or a consulting firm is selected in order to assist in achieving the goals of the following phases so as to increase organizational productivity and avoid a lack of continuous points [52];
3. Implementation: This phase addresses the exercise of installing and configuring the ESs solution to act upon the predefined vertical business needs of the organization.

Typically, this stage is owned by vendors and/or consulting firms that help in providing implementation best practices, methodologies, know-how, and staff training toward reaching the go-live milestone [53–55];

4. Use and Maintenance: Following the go-live milestone and with a pretext of understanding the aspects of ESs functionality, usability, and adequacy, trained staff begin using the system. Parallely, maintenance and system optimization through malfunction rectification is encountered [56,57];
5. Evolution: This phase corresponds to extending the capabilities of the implemented ES to deliver new use advantages. This is performed by broadening the vertical business functionalities to cover lateral functions across the organization;
6. Retirement: This phase discusses the stage of reaching either the functionality plateau through the organic growth of business needs and higher affinity toward technology adoption by the organization. In several cases, this stage is reached due to the counter-argument of an ES not reaching its utmost potential with the identification of other solutions that better address the business requirement of an organization [49].

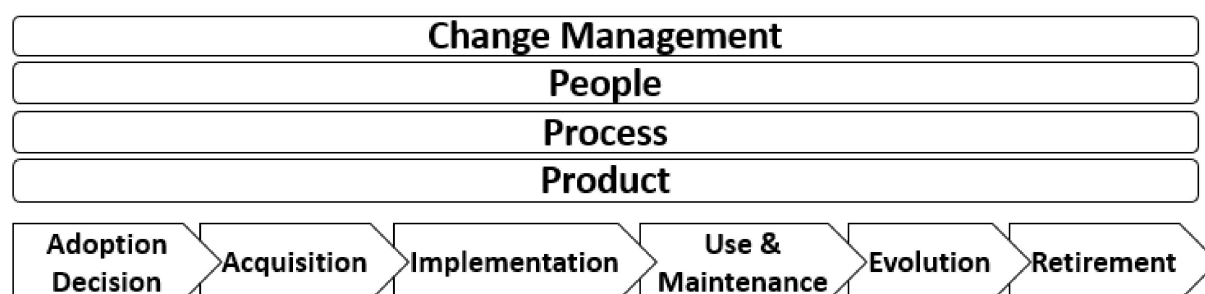


Figure 5. The ESs implementation lifecycle [5].

Despite that praised high values of ERPs and various ESs difficulties did not cease to exist. This has resulted in diminishing the sought-after return on investments (ROIs). ERP systems have been criticized for not maintaining the return on investments (ROIs) promised. Sykes et al. (2014) claim that 80% of ERP implementations fail. Moreover, 90% of large companies implementing ERP systems failed in their first trial [58]. It has also been reported that between 50–75% of US firms experience some degree of failure. Additionally, 65% of executives believe ERP implementation has at least a moderate chance of hurting business [59]. Three-quarters of ERP projects are considered failures and many ERP projects ended catastrophically [60]. Failure rates are estimated to be as high as 50% of all ERP implementations [61]. In addition, as much as 70% of ERP implementations fail to deliver anticipated benefits [62]. Still many ERP systems still face resistance and ultimately failure [63]. That been said, there are two main critical phases that might lead to the failure of an ERP project; the implementation and the post-implementation phases mark the two main phases in a given ERP lifecycle where many of the organizations might experience failure. The two phases include similar activities and involve similar stakeholders. Correspondingly, former research has identified certain critical success factors that are important for gaining benefits in organizations implementing ERP systems [64]. However, the cost of an unsuccessful ERP implementation can be high given the number of risks ERP projects experience [65,66].

3. Research Methodology

In order to fabricate the research skeleton, we have adopted the proposed literature review methodological framework by von Brocke [67], which is explained by the following steps: (1) definition of review scope, (2) conceptualization of the topic, (3) literature search, (4) literature analysis and synthesis, and (5) research agenda. The following subsections explain the steps taken in the literature review process following these steps.

On the other hand, to justify the outcomes from our investigative exercise further, the literature review taxonomy illustrated in Figure 6 is adopted; key taxonomy categories that are relevant to our exercise are highlighted in grey. This taxonomy had been initially devised by Cooper [68] and later featured by Randolph [69] as part of discussions guidelines discussions toward producing literature reviews.

Characteristic		Categories			
1	focus	research outcomes	research methods	theories	applications
2	goal	integration	criticism		central issues
3	organization	historical	conceptual		methodological
4	perspective	neutral representation		espousal of position	
5	audience	specialised scholars	general scholars	practitioners/politicians	general public
6	coverage	exhaustive	exhaustive and selective	representative	central/pivotal

Figure 6. Our literature review taxonomy, as proposed by Randolph [69].

3.1. Research Questions

By tying the earlier discussed challenges of deep learning and ESs that organizations face during implementations, the outlook for our current literature review is to inquire the following offspring research questions. These main research questions then set the compass toward investigating inquiring the scholar literature toward identifying the main ESs challenges landscape across the ESs implementation lifecycle for which deep learning technologies would be relevant. The definition of these research questions also defines the scholar repositories and focus academic outlets to be inquired as part of this review exercise. Correspondingly, these questions are:

- What are the ESs implementation lifecycle challenges being faced by organizations?
- What challenges could deep learning address these challenges? How?

3.2. Defining the Scope

By following the earlier explained framework and taxonomy we begin by defining the research scope. First, we begin by stipulating the focus to discuss the theoretical foundations of how the intervention of deep learning would be beneficial to address ESs challenges across the implementation lifecycle. This then sets the compass to set the literature search query criteria to a basket of top eight information systems journals [70]. Second, the formulation of the goal discusses the capabilities of deep learning versus the to be identified lifecycle challenges as being the central issues appearing through the analysis and synthesis review literature. Third, our article looks at the literature organization from a conceptual perspective to understand and organize the study to examine the theories in the literature discussing ESs challenges. Fourth, we adopt a neutral representation stance of the research mix by observing the cause and effect of the ESs challenges across the ESs implementation lifecycle as illustrated in Figure 5. The intended target audience of this article is both specialized scholars and practitioners within both fields of ESs and deep learning. Correspondingly, we conduct an exhaustive and selective citation coverage strategy across the specified journal basket.

3.3. Topic Conceptualization and Literature Search: Implementation Lifecycle Challenges

To instigate the conceptualization of our research and to establish an understanding of the different ESs challenges observed from the literature, we investigate the challenge of

diversity across the implementation lifecycle, which in turn constitutes the primary focus of the review. In parallel, we probe the literature posturing DL technologies within the context of ESs use.

The primary part formulates the research keywords used in the search process focusing on analyzing the scholarly topic of ESs challenges and its corresponding content to the point of convergence that is to be later overlaid on top of the ESs implementation lifecycle. The key terms used in the search were {"enterprise systems" + challenges} on Google Scholar, Web of Science, and Scopus across the eight basket journals. The range of search, on the other hand, encompassed an exhaustive article search with selective relevance toward the {"enterprise systems" + challenges} topic for a year range of six years (a period spanning the year 2015 to 2020).

The secondary part of the review analyzes how DL technologies had contributed toward an amalgamation with ESs literature devising solutions to some of the challenges experienced within the domain. The key terms used in the search were {"enterprise systems" + "deep learning"} on Google Scholar, Web of Science, and Scopus across the eight basket journals. The range of search, on the other hand, encompassed an exhaustive article search with selective relevance toward the {"enterprise systems" + "deep learning"} topic for the same year range of six years (a period spanning the year 2015 to 2020).

Table 2 showcases the number of record hits as extracted from Google Scholar, Web of Science, and Scopus versus the search criteria of {"enterprise systems" + challenges} for the period between 2015–2020 across the eight basket journals of AIS:

Table 2. Returned search hit counts with keyword {"enterprise systems" + challenges} between 2015 and 2020 for the eight basket journal outlets queried on Google Scholar, Web of Science, and Scopus.

Keywords: {"enterprise systems" + challenges} Range: 2015–2020	Scholar Repository	Total Hits
	Google Scholar	263
	Web of Science	61
	Scopus	32
	Total	356

On the other hand, Table 3 showcases the number of record hits as extracted from Google Scholar, Web of Science, and Scopus versus the search criteria of {"enterprise systems" + "deep learning"} for the period between 2015 and 2020 across the eight basket journals of AIS:

Table 3. Returned search hit counts with keyword {"enterprise systems" + "deep learning"} between 2015 and 2020 for the eight basket journal outlets queried on Google Scholar, Web of Science, Scopus.

Keywords: {"enterprise systems" + "deep learning"} Range: 2015–2020	Scholar Repository	Total Hits
	Google Scholar	7
	Web of Science	0
	Scopus	0
	Total	7

The articles exploited in the analysis and synthesis process were elected by the exclusion process based on an assessment of the abstracts and then the content. We had correspondingly triggered eight different search queries on Google Scholar to return corresponding hits while filtering by the journal names. Table 4 illustrates the search hits and relevance projection from the queried journal repositories, which formulates the scope of our reviewed articles as a combination of both search criteria illustrated above.

Table 4. Combined returned search hit counts with keyword {"enterprise systems" + challenges} and {"enterprise systems" + "deep learning"} between 2015 and 2020 for the eight basket journal outlets queried on Google Scholar, Web of Science, and Scopus.

Scholar Repositories: <ul style="list-style-type: none"> Google Scholar Web of Science Scopus Keywords: {"enterprise systems" + challenges} + {"enterprise systems" + "deep learning"} Range: 2013–2019	Journal Names	Total Hits	Included by Abstract	Included by Content
	Journal of AIS	52 + 1	11 + 1	9 + 1
	Information Systems Journal	37	7	6
	Information Systems Research	36 + 1	7 + 1	4 + 1
	Journal of Information Technology	81 + 2	12	7
	Journal of MIS	19	4	3
	Journal of SIS	27	6	4
	MIS Quarterly	55 + 1	12	8
	European Journal of IS	49 + 2	9	6
	Total	363	70	49
	Total Included		49	

All included the included/cited journal articles had been included for reference visual representational reference purposes in Appendix A. The corresponding topic focus across the ESs implementation lifecycle and the overlaying dimensions show precisely the pursued literature in relevance to the ESs implementation lifecycle. We had used this table to derive conclusions. The total count and percentages of total included articles are calculated for every phase and dimension within the implementation lifecycle. For clarity purposes, please see full details on the included/cited journal articles and the corresponding topic focus across the ESs implementation lifecycle in that table illustrated within the Table 1 of Appendix A. This was then summarized in the form of Table 5, which correspondingly helped with the analysis and synthesis arguments and set a compass toward the research agenda in the form of the directions to be discussed in the latter sections.

Table 5. The total count and percentages of total included articles for every phase and dimension within the ESs implementation lifecycle.

	Adoption	Acquisition	Implementation	Use and Maintenance	Evolution	Retirement	Change Management	People	Process	Product
Total Count	1	2	19	32	7	2	6	40	17	7
% of Total (out of 49 articles)	16%	4%	39%	65%	14%	4%	12%	82%	35%	7%

4. Analysis and Synthesis of Literature Search

The literature shows the tendencies across the ESs implementation lifecycle toward discussing ESs challenges experienced during the following phases: use and maintenance (65%), implementation (39%), adoption (16%), evolution (14%), retirement (4%), and acquisition (2%). On the other hand, the people dimension comes on top of the list (82%), followed by the process (35%), change management (12%), and product (7%).

The literature review illustrates the theoretical need to address the central issues identified across the implementation lifecycle for which deep learning architectures would be of benefit. This then paves the way for the research agenda setting the compass toward addressing how deep learning breakthroughs would tackle the uncovered implementation

phases and dimensions challenges. One of the main reasons why it was obvious that deep learning technologies would be of benefit for many of the identified challenges is because almost none of the reviewed literature did adopt any advanced analytics techniques to counter these effects. Correspondingly, the following sections highlight the topmost identified challenges within the reviewed literature set.

4.1. Implementation and Use and Maintenance Challenges

The tendency of reviewed literature to recite implementation and use and maintenance challenges had been evident. A summary of these top challenges is explained as follows:

- Problems with the customizations carried out during implementation:

This type of problem is one that is typically rooted during the implementation phase. No ESs solution is purposefully built to fit the business processes adopted by all organizations, bearing in mind the business practices variations from one business to the other. The solutions are usually packaged with best practice processes that are generic to work for a wide spectrum of organizations (even if the solution is an industry-focused solution). Given the granular details that distinguish one business practice from the other, unavoidable customization situations are experienced. This then puts the ESs on the track toward heavy customizations that derail the ESs solution completely from the intended design and value behind it.

- Problems with workarounds:

The recipe that is made of lack of training, the inability of users to comprehend the main purpose of a given ES, and the serious requirement posed by users to overcome business bottlenecks caused by the likes of this misunderstanding pushes employees to be creative about the use of the ESs solution. This also derails the system from the track of abiding by the best practices given solutions are packaged with. This type of problem could lead directly to the next problem given the inability to ensure validity, consistency, and integrity of the data being entered.

- Problems with data:

- This type comprises the inability of maintaining and enforcing validity, consistency, and integrity of the data being entered by the business users; ESs face tragic consequences given the inability of users to make efficient use of the systems. This then puts ESs on the course of requiring complementary solutions (like data quality and master data management solutions) to work on reactively and proactively remediating the data irregularities and imposing enterprise-wide data governance. However, many organizations do not easily come to the conclusion of requiring these solutions given these solutions and technologies are from an industrial standpoint still gaining eminence and typically are loosely defined, which creates confusion with information technology (IT) governance.

- Problems with support activities:

All the above problems then follow the natural streams of technical resources delving into the needs for support. This encompasses many different types of support streams and personas, from hiring external consultants to reverting back to vendor support teams assisting on solution bugs, glitches, and irregular behavior typically witnessed by the decision makers during the adoption and acquisition phases. The unfortunate truth is that this problem is generally witnessed through the dragging “ping-pong” effect across different parties, especially if one party has a pretext of hidden agendas. This opens many other doors knitted with controversy.

Upon adopting a new ES solution, many organizations tend to re-engineer the default processes and policies a given ES is delivered with. This work undertakes the process of altering the pre-packaged ES software to satisfy the functionality gap postured by the organizational business processes and policies [71]. This unfortunate practice poses an overwhelming impact on the latter rework related activities that are to be implemented

by the development team [72]. The literature explains that the amount of upgrade rework activities was directly proportional to the amount of customization a given ES undergoes. Many variants of these discussions had been handled across the literature set posturing a clear pattern and emphasis on the problems that customizations of ESs typically pose. ESs misfits then come to life despite the fact that many organizations are aware that customization practices typically cause implementation inefficiencies and usually ESs failure [13]. Adapting the processes to the ESs package to achieve the customizations typically promises a higher return on investment (ROI) that is not empirically solid. Customizations are usually affiliated with feature extensions for enhanced ESs use [67–73]. This practice of customizations poses a significant level of misunderstanding between the management and staff due to the lack of consensus on the goals and objectives that a given ES serves (perceived technology customization), with corresponding impacts on the organization's agility toward the business complexities [74,75]. All of this then leads to high risk and a major impact on post-implementation (use and maintenance, evolution and retirement) roles and functions [76].

On the other hand, workarounds did shape a significant part of the reviewed literature with noteworthy implications during the use and maintenance phase. This had also contributed to substantial process impediments caused by and to various roles (people) within the organization. The materialization of workarounds typically indicates that organizations still experience a lack of user satisfaction, one of the most intense researched fields in the area of IS research [77]. In situations where users experience IS activity slowness or obstruction, they tend to devise workarounds that help them reestablish control, seeking their business autonomy rather than exploring the system features [78,79]. The very aspect of adopting workaround practices commonly leads to data problems and inconsistencies within an ES solution [80,81]. It could be observed as a consequence that organizations naturally encounter IT compliance obstacles as a result of these practices. User resistance is also a very important topic handled because of business obstruction [82,83]. It is however important to mention that in a number of cases workarounds had been observed to be a method of lowering resistance to problematic implementations [13,77,83]. That being said, it could be observed that clubbing customizations and workaround practices inflame ES problems, leading to the eventual fate of ES system discontinuance and/or retirement [84].

4.2. The Socio-Technical and Compliance Challenges

The earlier section showcases how people and process dimensions of the lifecycle stand hand in hand in the line of ESs challenge fire along with the implementation phases. It had been evident how user resistance caused by workaround practices had contributed to the eventual discontinuance of given ESs. Psychological contract breaches (PCBs) had been used to further understand employees' attitudes and behavior. PCBs are typically driven by three dimensions of individuals' perceptions—reneging, congruence, and vigilance. The perception of a PCB regularly reduces employees' trust, job satisfaction, intention to remain with the organization, sense of obligation, and in-role and extra-role performance [85]. This then makes IS alignment within an organization a far-fetched goal [86]. Naturally, it could also be seen that one of the factors impacting IS alignment is the fact that routine use (RU) of a given ES is impacted highly by the perceived usefulness (PU) of the ES. It is also argued that PU is the most representative intrinsic motivator for IS use. Now, this very fact impacts significantly how innovative use (INV) is to be reached to build a consensus on acceptance of an IS. Similarly, there is a thin line between acceptance and resistance toward a given ES solution, which is typically defined on how what is known as a swift response phase is utilized [87]. Making the best use of swift response phases typically holds a significant weight toward setting the compass of PU from a user's point of view, which then defines the aptitude for an ES to either succeed or fail. Correspondingly, strategic IS alignment deems to pose an essential reliance on system use as a measure toward avoiding strategy blindness. This is where organizations are incapable of realizing the strategic intent of the implemented available system capabilities.

Certainly, the help of a strategic activity framework would come in handy to model how an ES is being used to support an emerging organizational strategy. This is where the ability to change system use as needs change (fluidity). In order to facilitate expedite the aspect of system fluidity a key people dimension needed to be visited; this is where champions are being elected within the organizational community to orient the rest of the organizational population toward system goals and processes and stipulate guided support to the people and process dimensions [76,78,87,88]. This is how various support structures could be complemented within an organization to enhance the impact of employee outcomes and hence effective IS investments [89]. Leadership empowerment plays an equally important role to ensure the effectiveness of IS investments and fluidity by adopting the planting the champion-based tactic. This plays a complementary essential role toward reducing developer stress safeguarding the entire ecosystem from the overwhelming collapse of support impacting the user PU [72]. This correspondingly defines how social capital could be reinforced to ensure the success of complex, cross-functional implementations [90].

Despite the fact that a number of frameworks and models had been devised in order to maneuver around the socio-technical aspects impacting the people and process dimensions of an ES implementation cycle, the counter fact of ensuring and reinforcing the adoption and application of these frameworks remains a challenge to date. These collective facts then onset the following research agendas.

4.3. Deep Learning for Enterprise Systems

Evidently from the reviewed intersecting literature, the infiltration of the likes of deep learning technologies in support for ESs shows a significant lack of pursuance. As illustrated from the search hits from three different repositories (Google Scholar, Web of Science, and Scopus) across the basket of eight AIS journals, the penetration of deep learning technologies to complement ESs is not getting much attention. Despite being hinted by Gregor [91] that automation introduced by non-human “automata” would contribute toward the overall algorithmic and deterministic nature of the system, an inclination toward human actors still stands. This is explained by the fact that human actors lean toward being more flexible in nature in achieving a certain goal. These very remarks to a great extent support our findings around workarounds where human actors deplete the algorithmic nature of ESs in many shapes and forms.

It was still argued in other instances that the intelligence factor introduced by machine learning, deep learning, and artificial intelligence alike can be of significant support for employee productivity. This was where the indicated benefits of the complimentary role of analytics technologies establish a foundation for enhanced productivity of all resources within an organization at large [92]. Again, that serves as clear support in favor of our hypothesis.

That all being said, we take on these findings into action to showcase the high accuracy levels deep learning technologies would be able to achieve as a step toward pursuing an evasion point from what is known as the point of “lack of continuous,” also known as “productivity decline,” within an ES lifecycle. This then contributes to the research directions to be discussed at a later stage of this scholarly work.

5. Experiment: Deep Learning for Augmented Technical Support

To assess the aptitude for deep learning solutions, a proof-of-concept experiment is to be conducted to showcase accuracy levels that could be achieved by the technology at hand. The accuracy levels achieved from the deep learning technology, which are to be illustrated, will consequently contribute toward the resolution of the above challenges. The foundational hypothesis, in this case, revolves around how deep learning technologies that are to be addressed would be able to process vast amounts of text representations within the feature space to derive text similarities between free text corpora in support tickets without compromising on the accuracy levels processed across the text feature space at

hand. This would then deliver evidence as to how deep learning could be of significant benefit in tackling the above challenges while maintaining the sought-after accuracy levels that qualify whether the solution would be of use or not.

Correspondingly, the purpose of this experiment is to illustrate the aptitude for deep learning algorithms to reach out to a level of ESs augmentation. In this example, we focus on one of the earlier explained ESs challenges, that is, the problems with support activities. This formulates an example as to how deep learning algorithms would be able to support a complete ES augmentation artifact. In this example, we have reused the code developed by Mueller [40]. The purpose of this exercise is to show how support ticket sentences and product documentation pairs would be able to be used as training data set for a Siamese-LSTM algorithm. It is to be noted that the reason why the Siamese LSTM architecture qualified to address the ESs augmentation at hand is given the architecture's ability to act on deriving complex text relationships throughout the feature space among the text embeddings guided by the training data sets. This distinguishes the architecture for traditional algorithms such as bag-of-words and Term Frequency-Inverse Document Frequency (TF-IDF) models [40]. That being said, the scored model of this algorithm would then be able to recommend useful resolutions as derived from the training data set.

5.1. Setup and Infrastructure

For the purpose of conducting this end-to-end experiment, the following Google cloud-based architecture was adopted to furnish the necessary foundation to conduct this experiment, serving the purpose of proving the concept. This was comprised four servers as depicted in the below diagram (Figure 7).

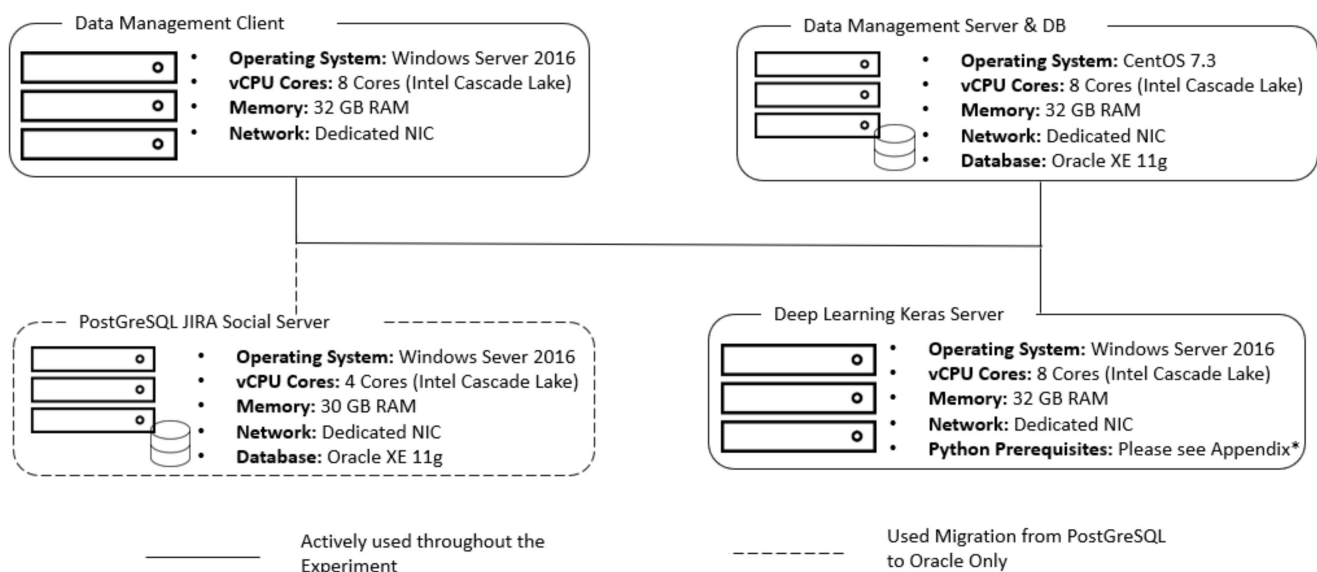


Figure 7. Infrastructure setup.

5.2. Data Acquisition and Preparation

It is widely known that issue tracking systems store valuable data examination assets that could be used for the purpose of testing hypotheses such as the one at hand. These data assets furnish reliable testing foundations that relate to the use and maintenance of ESs among which also many other dimensions could be analyzed to examine the socio-technical fronts. For the purpose of this experiment, we focused on aptitude for DL algorithms contributing to ESs augmentation.

It is important to note that acquiring similar data assets from real-life organizations had posed significant research challenges as it is with many of the scholars addressing the domain. To overcome this major hurdle, the Jira repository dataset as extracted by Ortu et.al. [93]. This dataset delivers a rather comprehensive collection of IT tickets from

the Jira issue tracking system (ITS) that spans many popular open-source ecosystems (Apache Software Foundation, Spring, JBoss, Hibernate Org., and CodeHaus). This dataset provides an abundance of data, serving as a comprehensive test bed across more than 1000 projects and containing more than 700,000 issue reports and more than 2 million issue comments across the four open-source ecosystems. Appendix D show the link for the GitHub repository shared by Ortu et.al. [93].

Despite the fact that the JIRA ITS repository serves as a gold mine of information that could be used for an experiment such as ours, the database relied on PostgreSQL as a container for the data to be shared across with the research community. This in turn meant that a level of inflexibility is attached given the reliance on this technology. To elevate the level of manageability and robustness, we had migrated the repository to an Oracle database to support the lifecycle of the experiment (please see Appendix B for the links to the Github repositories of both the JIRA Social Oracle and the PostgreSQL databases).

Throughout the process of data preparation, we have used an enterprise-grade data management tool that helped significantly with the complex steps of preparing the data for the Siamese LSTM experiment. As usual, the process of data preparation had consumed a significant amount of time to have the data ready for the model training, yet the use of a data management/integration tool helped expedite significantly. The fact that this data management tool did not support PostgreSQL was an additional reason why migration to a database like Oracle was necessary (please also see Appendix D containing the link to our GitHub repository for the migrated Oracle database). This is postured by the open-source nature of PostgreSQL databases and the rather typical low adoption levels on an enterprise level.

5.3. Training the Siamese LSTM Architecture

The process of running the Siamese LSTM network had been conducted over the full scope of the JIRA ITS repository while segregating the data into several datasets representing support tickets from different ecosystem communities. The python libraries used to setup the foundation for the Siamese LSTM model on Keras is depicted in Appendix C (Figure A2). This then translated into our ability to evaluate the outcome of the models and consistency across these different datasets representing the different communities. Conceptually, it was also quite important to ensure that contextual and latent features within the data are not distorted, which might have impacted representational forms of the similarities to be identified by the Siamese LSTM architecture. This step of segregation ensured the integrity of the constructed vector space drawing relations amongst the vast number of word embeddings. Table 6 showcases the comment count being used for training the model at hand.

Table 6. A record count of the free-text comments being used to train the Siamese LSTM architecture.

Repository	Free Text Comment Count
JBoss	387,285
CodeHaus	293,795
Spring	81,376
Hibernate	37,545
Total	800,001

5.4. Model Results

The in-use Siamese LSTM algorithm takes “word2vec” word embeddings to illustrate the latent semantic relatedness between given sentence pairs using Manhattan distances. This serves as the foundation to derive the semantic relatedness universe (represented by the vector space) used as the baseline of qualifying relations between technical problems and corresponding documented technical resolutions. In this case, the prepared training

data for the different ecosystem repositories had been split into 70/30 split between training and validation data. The training of the Siamese LSTM network had been done over 50 epochs. It must be noted that the usual time taken to train the network on CPUs has been significantly time-consuming for which Graphical Processing Units (GPUs) are always advisable given the multi-dimensional parallel computing. The following graphs depict the results of the accuracy and loss graphs conducted over a run of 50 epochs across the multiple runs of the algorithm across the four different ecosystem tickets.

As depicted in Figure 8, the example of the deep learning algorithm adopted in this experiment delivers reliable evidence through which the earlier discussed foundational hypothesis is being addressed. It is now quite visible how Siamese LSTM was able to process vast amounts of text representations within the feature space across a significant number of work embeddings. This then shows the high aptitude for such technology to derive text similarities between text corpora within feature space constituted out of support ticket repositories without compromising on the accuracy levels. Correspondingly, it could be concluded that deep learning technologies pose a good fit for augmenting ESs to cater to the challenges discussed observed as a result of the literature review discussed within this paper. This also converges with earlier scholarly work that highlights the vital role analytics-powered enterprise systems play within organizations suffering from ESs productivity decline [50].

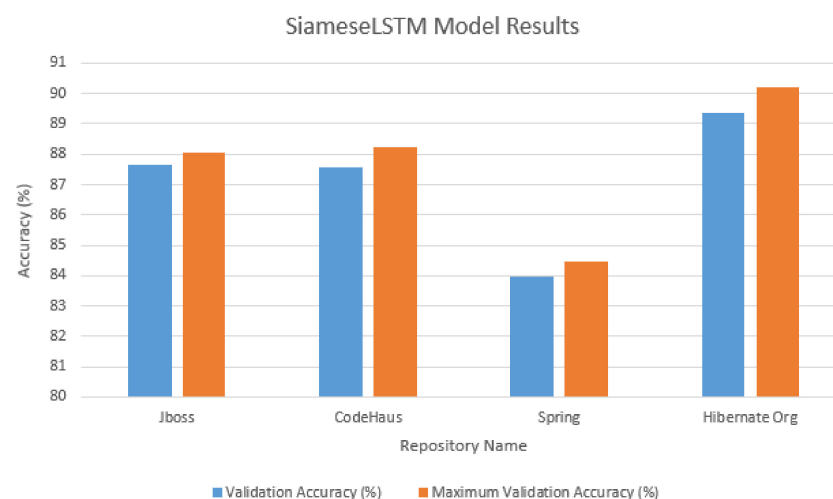


Figure 8. Model results from the deep learning model (Siamese LSTM) used as part of the conducted experiment across the four JIRA issue tracking repositories. Please see Appendix B for the detailed graphs produced by the model upon training for 50 epochs. Full model accuracy and loss versus the four handled issue tracking system (ITS) repositories are also illustrated on Figure A1 of Appendix B.

6. Research Agenda

The product of the earlier explained challenges drives several research agenda investigations that need to be interrogated. The most obvious of all are three main broad investigative directions. The first direction asks how DL technologies could be of applied benefit for ESs implementation phases, the second direction interrogates how DL technologies would empower the socio-technical landscape within an implementation lifecycle to reinforce organizational governance and compliance of processes and increase perceived usefulness, and the third direction probes how DL algorithms would change the de facto anatomy of an ES and whether it would necessitate an update to the definition of ESs or pose the rise of new breed of augmented/intelligent ESs.

The above broad queries are initially posed by the fact that more than 80% of information within an organization lies within unstructured formats. Many organizations abandon their information assets to be laid in the form of storage liabilities that only cost resources to store with no obvious use [77].

6.1. Direction 1: DL Augmentation of ESs Implementation Phases

Among the top agenda questions that need to be interrogated is what are the key aspects that DL augmentation on top of an ES? Hence, the following question is addressed:

- How can DL algorithms achieve a level of guided implementations through a level of augmentation?

As explained earlier, the ESs implementation lifecycle stipulates undergoing several implementation phases, as illustrated in Figure 1. It all begins with the adoption and acquisition phases where the organization decides on the ESs to be implemented. It had been evident from the literature that there had been little focus on the challenges encountered within these phases. This poses a stance toward investigating the corresponding challenges in these phases and maps an augmentation solution to guide the organization within these phases.

On the other hand, the concentrated focus on implementation and use and maintenance had uncovered many challenges ESs encounter, leading to either retirement of an ES or its replacement. The aptitude of DL technologies to address many of the highlighted challenges had been rarely highlighted in either direct or indirect ways, where some literature explained how advanced analytics algorithms including DL could serve the goal of safeguarding post-implementation phases [50]. This then addresses the problems of customizations, workarounds, and the augmentation of support functions. Despite this fact, more elaborate research is required to coagulate the hypothesis. On the other hand, the competitive capacity for DL algorithms to harness the 5Vs of big data poses an intriguing stance to cater to the data issues [42].

6.2. Direction 2: DL for Socio-Technical and Compliance Empowerment

The interaction between people and technology and their ability to comply with processes carries an important weight toward managing complex organizational work design set by ESs solutions. The earlier recited people and process challenges dictates a need for intelligent systems to support an organization to increase the perceived usefulness of an ES and promote rather innovative use approaches. This is when routine use could be accepted by users to avoid all that decapacitate ESs productivity, fluidity, and IS alignment that typically delivers a psychological contract breach effect [85].

The potential to counter these effects using DL algorithms then sets the stage for these intelligent algorithms to address the following by removing collaboration bottlenecks within various enterprise collaboration systems (ECs) caused by inefficient unstructured data use [94]:

- How DL algorithms could improve user compliance and increase the perceived usefulness of an ES?

The ability for DL algorithms to harness unstructured stored data within the organization and ESs solutions raise the queries on how these datasets could be intelligently re-channeled into the organization to empower the user compliance front. The propensity of DL techniques to empower an IT project compliance seems to outsize the socio-technical challenges. However, an adaptation of currently present frameworks, methodologies, and methods need to be put to the test to prove the hypothesis. The potential for DL to reinforce this front lies within enabling project custodians and champions to prove to top management that efficiencies are realized evidently with an impact on the practice and compliance of the system use. DL algorithms have the potential to be utilized for the purpose of intelligently advise users on best use and maintenance methods. This then paves the way into the following pointer:

- How DL algorithms could work on empowering the social capital within an organization?

Given technological aptitude for DL techniques to re-align compliant use within an organization, it would have the adeptness to enable the social capital within an organization. Likewise, this hypothesis calls for the scientific rigor to qualitatively assess the ability of the research body, especially around the area of how DL could augment the ESs users, support

functions, and executives to gain social capital. The aim here would be to catalyze the process of soliciting support from top management toward expanding on INV with a goal toward enhancing process compliance posed by the people. Through the ability for DL algorithms to analyze unstructured and semi-structured documents explained in Liu's research [90], a level of augmentation could be added to dissect structural, cognitive, and relational ties building the social capital. The tipping point of this research would be through a measure of how inclined would DL technologies management opinion could be influenced and, in many cases, turned.

6.3. Direction 3: The Importance of DL-ReFrams to Achieve ESs Augmentation

Despite the value that DL platforms deliver in the form of an augmented ESs, the complexities that emerge with these frameworks posture notable bottlenecks. That being said, the following inquiries need to be pursued:

- How can practice and research-based DL-ReFrams could be incubated in order to introduce de facto platforms that minimize the development efforts?

The need to continue the momentum toward establishing findable, accessible, interoperable, and reusable (FAIR) abiding frameworks that are easy to use postures significant importance for deep learning research. Given the exponential research inquiry rates of intelligent machine learning-based algorithms, the need toward sustaining the DL-ReFrams momentum becomes very important given the complexities postured by reusing intelligent machine learning algorithms for which, in our case, deep learning architectures come to be the main focus. It could be seen from our DL platforms section that various technology podiums (TensorFlow, Keras, etc.) document and pave the way toward significant advancements in the arena. However, only a handful of other podiums in the form of what we call DL-ReFrams are set toward increasing adoption and reusability. These are ones that, despite having a monumental impact, still suffer from the limited reach for spanning and addressing all types of use cases to come to an encyclopedic foundation to build on. This would correspondingly pose significant bottlenecks for interdisciplinary research streams that could benefit from the immense powers of the DL concept in its essence.

6.4. Direction 4: The Anatomy of ESs Technologies in Light of DL

Given the prominence of DL technologies to address all of the above, the status of weaving these approaches into the fabric of the so-called intelligent ESs requires inspection. This would query the ability to re-establish the de facto definition of what an intelligent ESs would be and the corresponding anatomical technology dissection. The reason behind this fact is that even though DL techniques possess immense power, they also come with challenges. Among some of the challenges is the aspect of incremental learning for non-stationary data. This is where streaming, fast-moving, and continuous data would require supplementary techniques to cater to those needs. Given the fact that streaming data typically come with noise, the ability to apply examples of denoising autoencoders (a variant of autoencoders) becomes crucial. Similarly, high dimensional data and large-scale models devised to address the explained augmentation onto ESs would pose enhancement requisites. Additionally, technological architecture build on deep learning techniques typically requires resource-intensive components to enable different phases of the learning process, especially during the training phase. Hence, the following queries are posed:

- What are the technical boundaries the might be encountered in this type of transformation?
- In what ways can the recited technical boundaries be avoided to facilitate the adoption using DL?
- Given the service-based offerings that ESs vendors provide, can the parsing issue be avoided by probably exposing Application Programming Interfaces (APIs) that add a level of augmented intelligence to the ESs to overcome the failure and ping-pong phenomenon?

6.5. Direction 5: Assessing the Aptitude of Transformers as Substitute to LSTM Architectures

Pushing the boundaries beyond the current capabilities of LSTM architectures, recent research had compared these architectures versus what is known as transformers [95]. Transformers are neural network architectures that focus on attention mechanisms to draw global dependencies without regard to the distance between the input or output sequences [96].

As shown in our experiment of LSTMs (a Siamese LSTM in our case), it could be quite difficult to train given the very long gradient paths that work their way throughout the vast number of word embeddings (i.e., an LSTM on a 100-word document typically would have gradients of close to a 100-layer network). The reason why a transformer network is faster because there is no need for the word embeddings to be ingested into the neural network architecture in sequence. The words can be passed into the transformer simultaneously given that there is no concept of the time step of the words being passed into the network architecture, hence, achieving a faster method sponsoring parallelism, which in turn postures a rather more efficient use of hardware. In turn, this lays a solid foundation that would be quite beneficial toward re-enforcing the very premise of ESs augmentation and the eventuality of transforming ESs architecture and state of the art.

7. Conclusions

The rise of intelligent applications had been evident through the emergence of deep learning architectures. Coming a long way from simple perceptrons, complex neural network architectures ascended to renovate the very essence of machine intelligence. In parallel, the ESs domain did see noteworthy research rigor studying many of the implementation lifecycle phases and dimensions. However, many of the ESs promises had experienced significant value depletion across the different phases of the implementation lifecycle given the misalignment of the people, process, product, and change management dimensions. The analysis of the reviewed literature indicates that the key phases seeing this depletion are primarily the implementation and use and maintenance phases. This conclusion had been reached by reviewing a basket of the top eight journals in the field of ESs.

This paper analyzed the most evident cases causing the effects of ESs value depletion, whereby both socio-technical and implementation lifecycle fronts were investigated throughout the review process and across the implementation lifecycle phases and dimensions. This lead to unearthing several research directions where the high aptitude of DL technologies could be postured for further investigation addressing the experienced depletion on both fronts. These questions are set the compass toward five directions—(1) how DL technologies could/would increase productivity throughout the implementation lifecycle, (2) how DL technologies could/would have an impact on socio-technical and compliance empowerment aspects within the influence context posed by ESs, (3) the importance of abiding by the FAIR principles to ensure streamlined and invigorated adoption of DL technologies through what we briefly introduced as DL-ReFrams, (4) how the key impact of this type of empowerment and augmentation would eventually redefine the ESs landscape, and (5) the aptitude for increasing the level of efficiencies using transformer-based architectures. These queries are postured versus the most evident review findings. Henceforth, the marriage of the most evident problems and DL capabilities lays the foundations for future research that would enable more efficient technology use, increased perceived usefulness of ESs, and the abolishment of the core challenges that ESs implementation lifecycles suffer from.

Correspondingly, the high impact of these findings tends to exhibit the industrial and academic potential toward redefining the de-facto ESs state-of-the-art. Academically, the progress toward the described directions sets the scholarly momentum toward laying the foundations for an “intelligent ES” as a new breed of applications and packaged business solutions. This should then lead and guide the industry into organically revamping the

ESs setup around the implementation lifecycle at large, may it be internal or external to both organizations and vendors.

Author Contributions: Writing—original draft, H.E.H.; Writing—review and editing, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://github.com/marcoortu/jira-social-repository> & <https://github.com/hassanien/ITSM-Oracle-DB-JIRA-Social-Repository>.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Included Literature Review Articles

Table 1. The included/cited journal articles and the corresponding topic focus across the ESs implementation lifecycle phases (highlighted in blue) XX and dimensions (highlighted in orange) marked in XX. It is to be noted that rows showing search hits coinciding between {“enterprise systems” + challenges} and {“enterprise systems” + “deep learning”} are highlighted in grey.

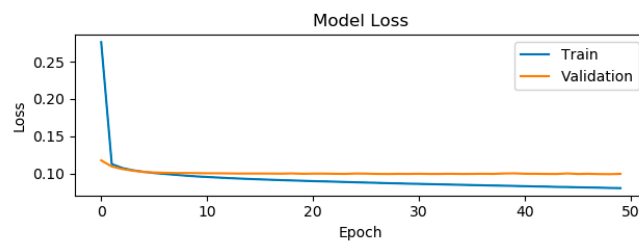
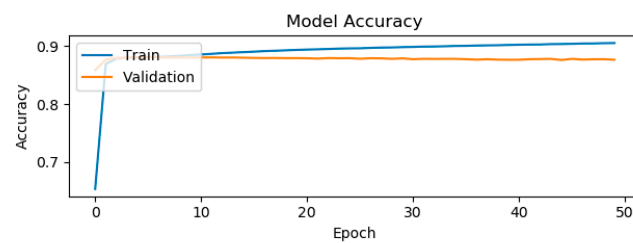
#	Outlet	Reference	Adoption	Acquisition	Implementat	Use & Maintenance	Evolution	Retirement	Change Management	People	Process	Product
1	JAIS	[78]			XX	XX				XX	XX	
2	JAIS	[97]				XX				XX		XX
3	JAIS	[90]			XX	XX				XX		XX
4	JAIS	[85]			XX	XX			XX	XX		
5	JAIS	[98]				XX	XX			XX		
6	JAIS	[88]				XX				XX	XX	
7	JAIS	[91]			XX					XX	XX	
8	JAIS	[99]			XX							XX
9	JAIS	[100]				XX				XX		
10	JAIS	[101]				XX	XX			XX		
11	ISJ	[79]	XX			XX				XX		
12	ISJ	[102]								XX		
13	ISJ	[13]			XX	XX					XX	
14	ISJ	[103]			XX	XX				XX		
15	ISJ	[81]				XX				XX	XX	
16	ISJ	[83]			XX	XX				XX	XX	XX
17	ISR	[87]				XX				XX	XX	
18	ISR	[72]			XX	XX				XX		
19	ISR	[104]	XX			XX						
20	ISR	[73]	XX			XX						XX
21	ISR	[92]			XX					XX		
22	JIT	[105]				XX				XX		
23	JIT	[71]			XX	XX				XX	XX	
24	JIT	[106]			XX				XX	XX		
25	JIT	[84]						XX		XX		
26	JIT	[74]				XX				XX		
27	JIT	[80]				XX			XX	XX	XX	
28	JIT	[107]	XX	XX	XX	XX	XX	XX	XX			XX
29	JMIS	[108]				XX				XX	XX	
30	JMIS	[109]	XX			XX				XX		
31	JMIS	[110]								XX	XX	XX
32	JSIS	[75]			XX						XX	
33	JSIS	[111]				XX				XX		

Table 1. Cont.

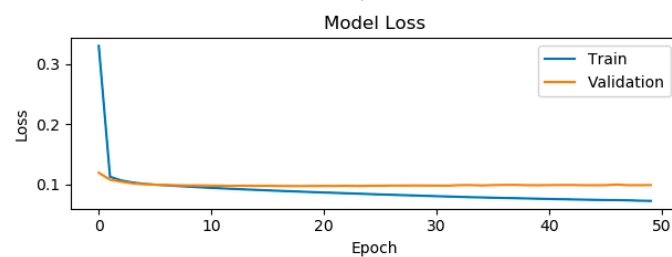
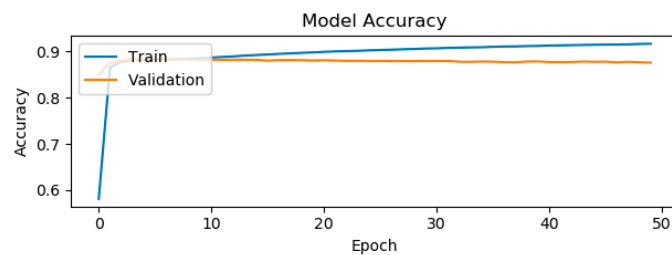
#	Outlet	Reference	Adoption	Acquisition	Implementat	Use & Maintenance	Evolution	Retirement	Change Management	People	Process	Product
34	JSIS	[112]					XX		XX	XX		
35	JSIS	[113]								XX	XX	
36	MISQ	[76]				XX				XX	XX	
37	MISQ	[89]			XX					XX		
38	MISQ	[114]	XX		XX					XX		
39	MISQ	[115]				XX				XX	XX	
40	MISQ	[116]				XX	XX			XX		
41	MISQ	[117]				XX				XX		
42	MISQ	[118]			XX	XX					XX	
43	MISQ	[119]				XX				XX		XX
44	EJIS	[120]	XX	XX	XX	XX	XX			XX		
45	EJIS	[82]			XX				XX	XX		
46	EJIS	[77]				XX				XX		
47	EJIS	[121]					XX					
48	EJIS	[122]	XX		XX					XX	XX	
49	EJIS	[86]									XX	

Appendix B

Siamese LSTM Model Results

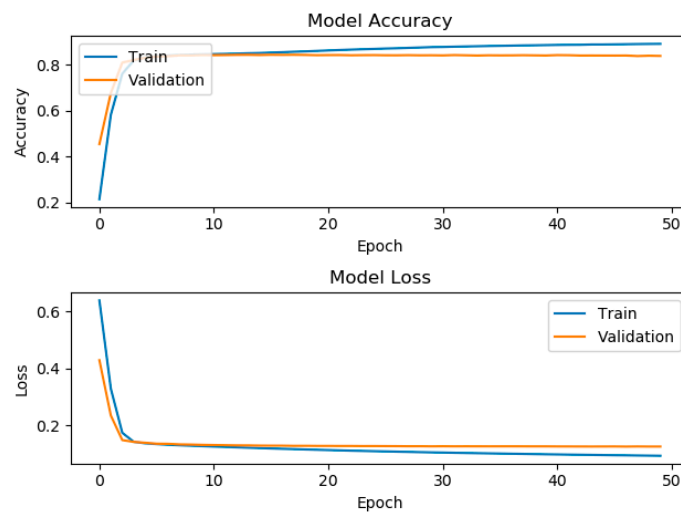


Training time finished: 50 epochs in 9989.57 s
Validation Accuracy: 0.8763(max: 0.8805)

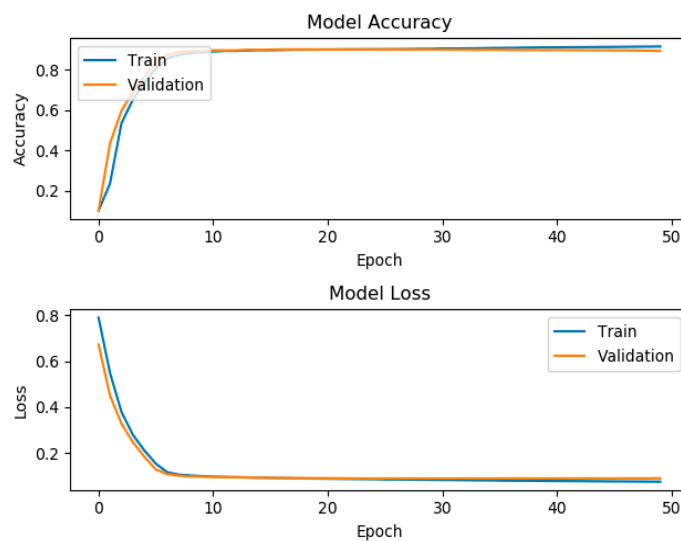


Training time finished: 50 epochs in 7475.87 s
Validation Accuracy: 0.8754(max: 0.8823)

Figure A1. Cont.



Training time finished: 50 epochs in 2061.56 s
Validation Accuracy: 0.8397(max: 0.8446)



Training time finished: 50 epochs in 1009.30 s
Validation Accuracy: 0.8934(max: 0.9019)

Figure A1. Siamese LSTM model accuracy and loss versus the four handled issue tracking system (ITS) repositories Sping, JBoss, CodeHaus, and Hibernate. (a) JBoss; (b) CodeHaus; (c) Spring; (d) ibernate Org.

Appendix C

Package	Version	Package	Version	Package	Version	Package	Version
Click	7.0	dill	0.3.0	mistune	0.8.4	pytz	2019.1
Flask	1.1.1	docutils	0.15.2	more-iter-tools	7.1.0	pywintpy	0.5.7
inja2	2.10.1	entrypoints	0.3	nbconvert	5.5.0	pyzmq	18.0.2
Keras	2.2.4	flake8	3.7.8	nbformat	4.4.0	qtconsole	4.5.1
Keras-Applications	1.0.8	flake8-docstrings	1.3.0	networkx	2.3	requests	2.22.0
Keras-Preprocessing	1.1.0	flake8-polyfill	1.0.2	nitk	3.4.5	s3transfer	0.3.3
Markdown	3.1.1	future	0.17.1	notebook	6.0.0	scikit-learn	0.22.1
MarkupSafe	1.1.1	gast	0.2.2	numpy	1.16.4	scipy	1.3.0
MatchZoo	2.1.0	gensim	3.8.1	packaging	19.0	setuptools	41.2.0
PyVAML	5.1.1	google-pasta	0.1.7	pandas	0.24.2	six	1.12.0
Pygments	2.4.2	grpcio	1.22.0	pandocfilters	1.4.2	sklearn	0.0
Send2Trash	1.5.0	h5py	2.9.0	parso	0.5.1	smart-open	1.9.0
Werkzeug	0.15.5	hyperopt	0.1.2	pexpect	4.7.0	snowballstemmer	1.9.0
absl-py	0.7.1	idna	2.8	pickleshare	0.7.5	tabulate	0.8.3
appnope	0.1.0	importlib-metadata	0.18	pip	19.2.3	tensorboard	1.14.0
astor	0.8.0	ipykernel	5.1.1	pluggy	0.12.0	tensorflow	1.14.0
atomicwrites	1.3.0	python	7.6.1	prometheus-client	0.7.1	tensorflow-estimator	1.14.0
attrs	19.1.0	python-gensim	0.2.0	prompt-toolkit	2.0.9	termcolor	1.1.0
backcall	0.1.0	pywidgets	7.5.0	protobuf	3.9.0	terminado	0.8.2
bleach	3.1.0	tsdangerous	1.1.0	ptyprocess	0.6.0	testpath	0.4.2
booto	2.40.0	yedi	0.14.1	py	1.8.0	tornado	6.0.3
botocore	1.12.1	mespath	0.9.4	pycodestyle	2.5.0	tgdm	4.32.2
botocore	1.15.1	joblib	0.14.1	pydocstyle	2.1.0	traitts	4.3.2
certifi	2019.11.28	jsonschema	3.0.1	pyflakes	2.1.1	urllib3	1.25.3
chardet	3.0.4	jupyter	1.0.0	pymongo	3.8.0	wcwidth	0.1.7
codecov	2.0.15	jupyter-client	5.3.1	pyarsing	2.4.0	webencodings	0.5.1
colorama	0.4.3	jupyter-console	6.0.0	pytest	0.15.3	wget	3.2
coverage	4.5.3	jupyter-core	4.5.0	pytest-cov	5.0.1	wheel	0.34.2
cycler	0.10.0	lxml	1.1.0	python-dateutil	2.8.0	widgetsnbestension	3.5.0
decorator	4.4.0	matplotlib	3.1.3	pytz	2019.1	wrapt	1.11.2
defusedxml	0.6.0	nccebe	0.6.1			yapp	0.5.2

Figure A2. Python-based Keras libraries and prerequisites used for the experiment.

The following is a depiction of the full packages and libraries being used for the purpose of setting up the python infrastructure to conduct the Siamese LSTM experiment.

Appendix D

JIRA Social Repositories

D1. JIRA Social Repository as delivered by Ortu et.al. [93] (PostgreSQL version)

Available online: <https://github.com/marcoortu/jira-social-repository> (Accessed on 17 February 2021)

D2. Migrated version of the JIRA Social Repository to an Oracle Database

Available online: <https://github.com/hhassanien/ITSM-Oracle-DB-JIRA-Social-Repository> (Accessed on 17 February 2021)

References

- Deng, L.; Yu, D. Deep Learning: Methods and Applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [CrossRef]
- Najafabadi, M.M.; Villanustre, F.; Khoshgoftaar, T.M.; Seliya, N.; Wald, R.; Muharemagic, E. Deep Learning applications and challenges in big data analytics. *J. Big Data* **2015**, *2*, 1. [CrossRef]
- Boza, A.; Cuenca, L.; Poler, R.; Michaelides, Z. The interoperability force in the ERP field. *Enterp. Inf. Syst.* **2015**, *9*, 257–278. [CrossRef]
- Markus, M.L.; Tanis, C. The Enterprise System Experience—From Adoption to Success. *Fram. Domains IT Res. Manag. Proj. Futur. Through Past* **2000**, *173*, 173–207.
- Esteves, J.M.J.; Pastor, J.A. An ERP Life-cycle-based Research Agenda. In Proceedings of the 1st International Workshop in Enterprise Management & Resource Planning, Venice, Italy, 25–26 November 1999; pp. 359–371.
- Markus, M.L.; Axline, S.; Petrie, D.; Tanis, C. Learning from adopters' experiences with ERP: Problems encountered and success achieved. *J. Inf. Technol.* **2000**, *15*, 245–265.
- Alter, S. Theory of Workarounds. *Commun. Assoc. Inf. Syst.* **2013**, *34*, 1–30. [CrossRef]
- Volkoff, O.; Strong, D.M.; Elmes, M.B. Technological Embeddedness and Organizational Change. *Organ. Sci.* **2007**, *18*, 832–848. [CrossRef]
- Lindley, J.T.; Topping, S.; Lindley, L.T. The hidden financial costs of ERP software. *Manag. Financ.* **2008**, *34*, 78–90. [CrossRef]
- Law, C.C.H.; Chen, C.C.; Wu, B.J.P. Managing the full ERP Life-Cycle: Considerations of maintenance and support requirements and IT governance practice as integral elements of the formula for successful ERP adoption. *Comput. Ind.* **2010**, *61*, 297–308. [CrossRef]
- Sarkis, J.; Sundarraj, R.P. Managing large-scale global enterprise resource planning systems: A case study at Texas Instruments. *Int. J. Inf. Manag.* **2003**, *23*, 431–442. [CrossRef]
- Hustad, E.; Bechina, A. A study of the ERP Project Life Cycles in Small-and-Medium-Sized Enterprises: Critical Issues and Lessons Learned. *World Acad. Sci. Eng. Technol.* **2011**, *5*, 85–91.

13. Van Beijsterveld, J.A.A.; Van Groenendaal, W.J.H. Solving misfits in ERP implementations by SMEs. *Inf. Syst. J.* **2016**, *26*, 369–393. [CrossRef]
14. Gibson, A.; Patterson, J. *Deep Learning: A Practitioner's Approach*; O'Reilly Media, Inc.: Newton, MA, USA, 2017; ISBN 978-1-491-91425-0.
15. Schmidhuber, J. Deep Learning. *Scholarpedia* **2015**, *10*. [CrossRef]
16. Tang, D.; Qin, B.; Liu, T. Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 292–303. [CrossRef]
17. Bengio, Y. *Learning Deep Architectures for AI*; Now Publishers Inc.: Boston, MA, USA, 2009; Volume 2, ISBN 978-1-60198-294-0.
18. Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; Voorhees, E.M. Overview of the TREC 2019 deep learning track. *arXiv*, 2020; arXiv:2003.07820.
19. LeCun, Y.; Bengio, Y.; Hinton, G.; Schmidhuber, J.; Rusk, N. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
20. Collobert, R.; Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
21. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
22. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, Bellevue, WA, USA, 2 July 2012; pp. 37–49.
23. Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.-A.; Vincent, P. Why Does Unsupervised Pre-Training Help Deep Learning? In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; Volume 11, pp. 201–208.
24. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef] [PubMed]
25. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
26. Bengio, Y.; LeCun, Y. Scaling Learning Algorithms towards AI. *Large Scale Kernel Mach.* **2007**, *34*, 1–41.
27. Hinton, G. A Practical Guide to Training Restricted Boltzmann Machines. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 599–619.
28. Sarikaya, R.; Hinton, G.E.; Deoras, A. Application of deep belief networks for natural language understanding. *IEEE Trans. Audio Speech Lang. Process.* **2014**, *22*, 778–784. [CrossRef]
29. Jiang, M.; Liang, Y.; Feng, X.; Fan, X.; Pei, Z.; Xue, Y.; Guan, R. Text classification based on deep belief network and softmax regression. *Neural Comput. Appl.* **2018**, *29*, 61–70. [CrossRef]
30. Zhang, X.; LeCun, Y. Text Understanding from Scratch. *arXiv* **2015**, arXiv:1502.01710.
31. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
32. Le, H.T.; Cerisara, C.; Denis, A. Do Convolutional Networks need to be Deep for Text Classification? *arXiv* **2018**, arXiv:1707.04108.
33. Schmidhuber, J.; Graves, A.; Gomez, F.; Hochreiter, S. *The Recurrent Neural Networks*; Cambridge University Press: Cambridge, UK, 2015; Available online: <https://people.idsia.ch/~juergen/rnnbook.html> (accessed on 11 December 2020).
34. Menaga, D.; Revathi, S. *Deep Learning: A Recent Computing Platform for Multimedia Information Retrieval*; IGI Global: Hershey, PA, USA, 2020; pp. 121–141. [CrossRef]
35. Guo, J.; Fan, Y.; Pang, L.; Yang, L.; Ai, Q.; Zamani, H.; Wu, C.; Croft, W.B.; Cheng, X. A Deep Look into neural ranking models for information retrieval. *Inf. Process. Manag.* **2020**, *57*, 102067. [CrossRef]
36. Li, H.; Lu, Z. Deep learning for information retrieval. In Proceedings of the 39th Conference on Research and Development in Information Retrieval, Pisa, Italy, 17–21 July 2016; pp. 1203–1206.
37. Goldberg, Y. *Neural Network Methods in Natural Language Processing*; Synthesis Lectures on Human Language Technologies 10.1. 2017. Available online: <https://doi.org/10.2200/S00762ED1V01Y201703HLT037> (accessed on 11 December 2020).
38. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
39. Allen, C.; Hospedales, T. Analogies explained: Towards understanding word embeddings. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 337–347.
40. Mueller, J.; Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2786–2792.
41. Hatcher, W.G.; Yu, W. A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends. *IEEE Access* **2018**, *6*, 24411–24432. [CrossRef]
42. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [CrossRef]
43. Jauro, F.; Chiroma, H.; Gital, A.Y.; Almutairi, M.; Abdulhamid, S.M.; Abawajy, J.H. Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend. *Appl. Soft Comput. J.* **2020**, *96*, 106582. [CrossRef]
44. Li, M.; Liu, Y.; Liu, X.; Sun, Q.; You, X.; Yang, H.; Luan, Z.; Gan, L.; Yang, G.; Qian, D. The Deep Learning Compiler: A Comprehensive Survey. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 708–727. [CrossRef]

45. Piatetsky-Shapiro, G. KDnuggets. Available online: <https://www.kdnuggets.com/> (accessed on 17 February 2021).
46. Comparison of Deep-Learning Software. Available online: https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software (accessed on 17 February 2021).
47. Fan, Y.; Pang, L.; Hou, J.; Guo, J.; Lan, Y.; Cheng, X. MatchZoo: A Toolkit for Deep Text Matching. *arXiv* **2017**, arXiv:1707.07270.
48. Molino, P.; Dudin, Y.; Miryala, S.S. Introducing Uber Ludwig, a Code Free Deep Learning Toolbox. Available online: <https://eng.uber.com/introducing-ludwig/> (accessed on 14 March 2020).
49. Haddara, M.; Elragal, A. ERP lifecycle: When to retire your ERP system? In Proceedings of the International Conference on ENTERprise Information Systems, Berlin/Heidelberg, Germany, 5–7 October 2011; pp. 168–177.
50. Elragal, A.; Hassanien, H.E. Augmenting Advanced Analytics into Enterprise Systems: A Focus on Post-Implementation Activities. *Systems* **2019**, *7*, 31. [\[CrossRef\]](#)
51. Chung, S.H.; Snyder, C.A. ERP adoption: A technological evolution approach. *Int. J. Agil. Manag. Syst.* **2000**, *2*, 24–32. [\[CrossRef\]](#)
52. Bhatti, T. Acquisition (Purchasing) of ERP systems from organizational buying behavior perspective. *Int. J. Bus. Soc. Res.* **2014**, *4*, 33–46.
53. Brown, W. Enterprise resource planning (ERP) implementation planning and structure: A recipe for ERP success. In Proceedings of the 32nd Annual ACM SIGUCCS Conference on User Services, Baltimore, MD, USA, 10–13 October 2004; pp. 82–86. [\[CrossRef\]](#)
54. Haines, M.N.; Goodhue, D.L. ERP Implementations: The Role of Implementation Partners and Knowledge Transfer. In Proceedings of the Institute of Rural Management Conference, Anchorage, AL, USA, 21–24 May 2000; pp. 34–38.
55. Nah, F.F.; Lau, J.L.; Kuang, J. Critical factors for successful implementation of enterprise systems. *Bus. Process Manag. J.* **2001**, *7*, 285–296.
56. Hecht, S.; Wittges, H.; Krcmar, H. IT capabilities in ERP maintenance—A review of the ERP post-implementation literature. In Proceedings of the European Conference on Information Systems, Helsinki, Finland, 9–11 June 2011.
57. Light, B. The maintenance implications of the customization of ERP software. *J. Softw. Maint. Evol.* **2001**, *13*, 415–429. [\[CrossRef\]](#)
58. Donovan, R.M. Successful ERP Implementation the First Time. *Perform. Improv.* **2001**, *25*, 1–3.
59. Umble, E.J.; Umble, M.M. Avoiding ERP implementation failure. *Ind. Manag.* **2000**, *44*, 25.
60. Rasmy, M.; Tharwat, A.; Ashraf, S. Enterprise resource planning (ERP) implementation in the Egyptian organizational context. In Proceedings of the European and Mediterranean Conference on Information Systems, Cairo, Egypt, 31 October–4 November 2005.
61. Muscatello, J.R.; Parente, D.H. Enterprise Resource Planning (ERP): A Postimplementation Cross-Case Analysis. *Inf. Resour. Manag. J.* **2006**, *19*, 61–80. [\[CrossRef\]](#)
62. Wang, E.T.G.; Chia-Lin Lin, C.; Jiang, J.J.; Klein, G. Improving enterprise resource planning (ERP) fit to organizational process through knowledge transfer. *Int. J. Inf. Manag.* **2007**, *27*, 200–212. [\[CrossRef\]](#)
63. Osnes, K.B.; Olsen, J.R.; Vassilakopoulou, P.; Hustad, E. ERP Systems in Multinational Enterprises: A literature Review of Post-implementation Challenges. *Procedia Comput. Sci.* **2018**, *138*, 541–548. [\[CrossRef\]](#)
64. Somers, T.M.; Nelson, K. The impact of critical success factors across the stages of enterprise resource planning implementations. In Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 3–6 January 2001; pp. 1–10.
65. Sun, A.Y.T.; Yazdani, A.; Overend, J.D. Achievement assessment for enterprise resource planning (ERP) system implementations based on critical success factors (CSFs). *Int. J. Prod. Econ.* **2005**, *98*, 189–203. [\[CrossRef\]](#)
66. Huang, S.-M.; Chang, I.-C.; Li, S.-H.; Lin, M.-T. Assessing risk in ERP projects: Identify and prioritize the factors. *Ind. Manag. Data Syst.* **2004**, *104*, 681–688. [\[CrossRef\]](#)
67. Von Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Plattfaut, R.; Cleven, A. Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In Proceedings of the 17th European Conference on Information Systems, Verona, Italy, 8–10 June 2009; Volume 9, pp. 2206–2217.
68. Cooper, H.M. Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowl. Soc.* **1988**, *1*, 104–126. [\[CrossRef\]](#)
69. Randolph, J.J. A Guide to Writing the Dissertation Literature Review. *Pract. Assess. Res. Eval.* **2009**, *14*, 1–13.
70. Association of Information Systems: Senior Scholars’ Basket of Journals. Available online: <https://aisnet.org/page/SeniorScholarBasket> (accessed on 17 February 2021).
71. Fryling, M. Investigating the Effect of Customization on Rework in a Higher Education Enterprise Resource Planning (ERP) Post-Implementation Environment: A System Dynamics Approach. *J. Inf. Technol. Case Appl. Res.* **2015**, *17*, 8–40. [\[CrossRef\]](#)
72. Windeler, J.B.; Maruping, L.; Venkatesh, V. Technical systems development risk factors: The role of empowering leadership in lowering developers’ stress. *Inf. Syst. Res.* **2017**, *28*, 775–796. [\[CrossRef\]](#)
73. Zhang, Z.; Nan, G.; Tan, Y. Cloud Services vs. On-Premises Software: Competition Under Security Risk and Product Customization. *Inf. Syst. Res.* **2020**, *31*, 848–864. [\[CrossRef\]](#)
74. Chadhar, M.; Daneshgar, F. Organizational Learning and ERP Post-implementation Phase: A Situated Learning Perspective. *J. Inf. Technol.* **2018**, *19*, 138–156.
75. Ravichandran, T. Exploring the relationships between IT competence, innovation capacity and organizational agility. *J. Strateg. Inf. Syst.* **2018**, *27*, 22–42. [\[CrossRef\]](#)
76. Tian, F.; Xu, S.X. How Do Enterprise Resource Planning Systems Affect Firm Risk? Post-Implementation Impact. *MIS Q.* **2015**, *39*, 39–60. [\[CrossRef\]](#)

77. Laumer, S.; Maier, C.; Weitzel, T. Information quality, user satisfaction, and the manifestation of workarounds: A qualitative and quantitative study of enterprise content management system users. *Eur. J. Inf. Syst.* **2017**, *26*, 333–360. [\[CrossRef\]](#)
78. Cram, W.A.; Brohman, M.K.; Gallupe, R.B. Information systems control: A review and framework for emerging information systems processes. *J. Assoc. Inf. Syst.* **2016**, *17*, 216–266. [\[CrossRef\]](#)
79. Bala, H.; Bhagwatwar, A. Employee dispositions to job and organization as antecedents and consequences of information systems use. *Inf. Syst. J.* **2018**, *28*, 650–683. [\[CrossRef\]](#)
80. Lederman, R.; Kurnia, S.; Peng, F.; Dreyfus, S. Tick a box, any box: A case study on the unintended consequences of system misuse in a hospital emergency department. *J. Inf. Technol. Teach. Cases* **2015**, *5*, 74–83. [\[CrossRef\]](#)
81. Deng, X.N.; Wang, T.; Galliers, R.D. More than providing “solutions”: Towards an understanding of customer-oriented citizenship behaviours of IS professionals. *Inf. Syst. J.* **2015**, *25*, 489–530. [\[CrossRef\]](#)
82. Laumer, S.; Maier, C.; Eckhardt, A.; Weitzel, T. Work routines as an object of resistance during information systems implementations: Theoretical foundation and empirical evidence. *Eur. J. Inf. Syst.* **2016**, *25*, 317–343. [\[CrossRef\]](#)
83. Malaurent, J.; Karanasios, S. Learning from Workaround Practices: The Challenge of Enterprise System Implementations in Multinational Corporations. *Inf. Syst. J.* **2020**, *30*, 639–663. [\[CrossRef\]](#)
84. Recker, J. Reasoning about Discontinuance of Information System Use. *J. Inf. Technol. Theory Appl.* **2016**, *17*, 41–66.
85. Lin, T.; Huang, S.; Chiang, S.-C. User Resistance to the Implementation of Information Systems: A Psychological Contract Breach Perspective. *J. Assoc. Inf. Syst.* **2018**, *19*, 306–332. [\[CrossRef\]](#)
86. Bygstad, B.; Øvrelid, E. Architectural alignment of process innovation and digital infrastructure in a high-tech hospital. *Eur. J. Inf. Syst.* **2020**, *29*, 220–237. [\[CrossRef\]](#)
87. Tong, Y.; Tan, S.S.-L.; Teo, H.-H. The Road to Early Success: Impact of System Use in the Swift Response Phase. *Inf. Syst. Res.* **2015**, *26*, 418–436. [\[CrossRef\]](#)
88. Morana, S. Designing Process Guidance Systems. *J. Assoc. Inf. Syst.* **2018**, *20*, 1–60. [\[CrossRef\]](#)
89. Sykes, T.A. Support Structures and Their Impacts on Employee Outcomes: A Longitudinal Field Study of an Enterprise System Implementation. *MIS Q.* **2015**, *39*, 473–495. [\[CrossRef\]](#)
90. Liu, G.; Wang, E.; Chua, C. Leveraging Social Capital to Obtain Top Management Support in Complex, Cross-Functional IT Projects. *J. Assoc. Inf. Syst.* **2018**, *16*, 707–737. [\[CrossRef\]](#)
91. Gregor, S.; Chandra Kruse, L.; Seidel, S. The Anatomy of a Design Principle. *J. Assoc. Inf. Syst.* **2020**, *21*, 1622–1652.
92. Gopal, A.; Karmegam, S.R.; Koka, B.R.; Rand, W.M. Is the grass greener? On the strategic implications of moving along the value chain for IT service providers. *Inf. Syst. Res.* **2020**, *31*, 148–175. [\[CrossRef\]](#)
93. Ortu, M.; Destefanis, G.; Adams, B.; Murgia, A.; Marchesi, M.; Tonelli, R. The JIRA repository dataset: Understanding social aspects of software development. In Proceedings of the 11th International Conference on Predictive Models and Data Analytics in Software Engineering, Beijing, China, 21 October 2015.
94. Prakash, S.; Joshi, S.; Bhatia, T.; Sharma, S.; Samadhiya, D.; Shah, R.R.; Kaiwartya, O.; Prasad, M. Characteristic of enterprise collaboration system and its implementation issues in business management. *Int. J. Bus. Intell. Data Min.* **2020**, *16*, 49–65. [\[CrossRef\]](#)
95. Zeyer, A.; Bahar, P.; Irie, K.; Schluter, R.; Ney, H. A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Sentosa, Singapore, 14–18 December 2019; pp. 8–15.
96. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
97. Kulkarni, U.; Robles-Flores, J.; Popovič, A. Business Intelligence Capability: The Effect of Top Management and the Mediating Roles of User Participation and Analytical Decision Making Orientation. *J. Assoc. Inf. Syst.* **2018**, *18*, 516–541. [\[CrossRef\]](#)
98. Recker, J.; Indulska, M.; Green, P.F.; Burton-Jones, A.; Weber, R. Information systems as representations: A review of the theory and evidence. *J. Assoc. Inf. Syst.* **2019**, *20*, 735–786. [\[CrossRef\]](#)
99. Widjaja, T.; Gregory, R.W. Monitoring the complexity of its architectures: Design principles and an its artifact. *J. Assoc. Inf. Syst.* **2020**, *21*, 664–694.
100. Austin Kwak, D.H.; Ma, X.; Polites, G.; Srite, M.; Hightower, R.; Haseman, W.D. Cross-level moderation of team cohesion in individuals’ utilitarian and hedonic information processing: Evidence in the context of team-based gamified training. *J. Assoc. Inf. Syst.* **2019**, *20*, 161–185.
101. Carter, M.; Petter, S.; Grover, V.; Thatcher, J.B. IT Identity: A Measure and Empirical Investigation of its Utility to IS Research. *J. Assoc. Inf. Syst.* **2020**, *21*, 2.
102. Shollo, A.; Galliers, R.D. Towards an understanding of the role of business intelligence systems in organisational knowing. *Inf. Syst. J.* **2016**, *26*, 339–367. [\[CrossRef\]](#)
103. Seddon, P.B.; Constantinidis, D.; Tamm, T.; Dod, H. How does business analytics contribute to business value? *Inf. Syst. J.* **2017**, *27*, 237–269. [\[CrossRef\]](#)
104. Greenwood, B.N.; Ganju, K.K.; Angst, C.M. How does the implementation of enterprise information systems affect a professional’s mobility? An empirical study. *Inf. Syst. Res.* **2019**, *30*, 563–594. [\[CrossRef\]](#)
105. Mettler, T.; Winter, R. Are business users social? A design experiment exploring information sharing in enterprise social systems. *J. Inf. Technol.* **2016**, *31*, 101–114. [\[CrossRef\]](#)

106. Grainger, N.; McKay, J. The long and winding road of enterprise system implementation: Finding success or failure? *J. Inf. Technol. Teach. Cases* **2015**, *5*, 92–101. [[CrossRef](#)]
107. Schneider, S.; Wollersheim, J.; Krcmar, H.; Sunyaev, A. How do requirements evolve over time? A case study investigating the role of context and experiences in the evolution of enterprise software requirements. *J. Inf. Technol.* **2018**, *33*, 151–170. [[CrossRef](#)]
108. Müller, O.; Fay, M.; vom Brocke, J. The Effect of Big Data and Analytics on Firm Performance: An Econometric Analysis Considering Industry Characteristics. *J. Manag. Inf. Syst.* **2018**, *35*, 488–509. [[CrossRef](#)]
109. Lai, V.S.; Lai, F.; Lowry, P.B. Technology Evaluation and Imitation: Do They Have Differential or Dichotomous Effects on ERP Adoption and Assimilation in China? *J. Manag. Inf. Syst.* **2016**, *33*, 1209–1251. [[CrossRef](#)]
110. Cui, T.; Tong, Y.; Teo, H.H.; Li, J. Managing Knowledge Distance: IT-Enabled Inter-Firm Knowledge Capabilities in Collaborative Innovation. *J. Manag. Inf. Syst.* **2020**, *37*, 217–250. [[CrossRef](#)]
111. Rossi, M.; Nandhakumar, J.; Mattila, M. Balancing fluid and cemented routines in a digital workplace. *J. Strateg. Inf. Syst.* **2020**, *29*, 101616. [[CrossRef](#)]
112. Baptista, J.; Stein, M.K.; Klein, S.; Watson-Manheim, M.B.; Lee, J. Digital work and organisational transformation: Emergent Digital/Human work configurations in modern organisations. *J. Strateg. Inf. Syst.* **2020**, *29*, 101618. [[CrossRef](#)]
113. Koo, Y.; Park, Y.K.; Ham, J.; Lee, J.N. Congruent patterns of outsourcing capabilities: A bilateral perspective. *J. Strateg. Inf. Syst.* **2019**, *28*, 101580. [[CrossRef](#)]
114. Baesens, B.; Bapna, R.; Marsden, J.R.; Vanthienen, J.; Zhao, J.L. Transformational Issues of Big Data and Analytics in Networked Business. *MIS Q.* **2016**, *40*, 807–818. [[CrossRef](#)]
115. Gust, G.; Flath, C.M.; Brandt, T.; Strohle, P.; Neumann, D. How a Traditional Company Seeded New Analytics Capabilities. *MIS Q. Exec.* **2017**, *16*, 10–13.
116. Mocker, M.; Boochever, J.O. How to avoid enterprise systems landscape complexity. *MIS Q. Exec.* **2020**, *19*, 57–68. [[CrossRef](#)]
117. Lauterbach, J.; Mueller, B.; Kahrau, F.; Maedche, A. Achieving Effective Use When Digitalizing Work: The Role of Representational Complexity. *Rev. Lit. Arts Am.* **2020**, *44*, 1023–1048.
118. Venkatesh, V.; Sykes, T.A.; Rai, A.; Setia, P. Governance and ICT4D initiative success: A longitudinal field study of ten villages in rural India. *MIS Q. Manag. Inf. Syst.* **2019**, *43*, 1081–1104.
119. Setia, P.; Bayus, B.L.; Rajagopalan, B. The Takeoff of Open Source Software: A Signaling Perspective Based on Community Activities. *MIS Q. Manag. Inf. Syst.* **2020**, *44*, 1439–1458. [[CrossRef](#)]
120. Shao, Z.; Feng, Y.; Hu, Q. Effectiveness of top management support in enterprise systems success: A contingency perspective of fit between leadership style and system life-cycle. *Eur. J. Inf. Syst.* **2016**, *25*, 131–153. [[CrossRef](#)]
121. Hedman, J.; Sarker, S. Information system integration in mergers and acquisitions: Research ahead. *Eur. J. Inf. Syst.* **2015**, *24*, 117–120. [[CrossRef](#)]
122. Bala, H.; Hossain, M.M.; Bhagwatwar, A.; Feng, X. Ownership and governance, scope, and empowerment: How does context affect enterprise systems implementation in organisations in the Arab World? *Eur. J. Inf. Syst.* **2020**, 1–27. [[CrossRef](#)]