IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 7, 2022, accepted March 20, 2022, date of publication April 12, 2022, date of current version April 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3166628

A Cognitive Deception Model for Generating Fake Documents to Curb Data Exfiltration in Networks During Cyber-Attacks

OLAYIWOLA TOKUNBO TAOFEEK[1], MOATSUM ALAWIDA[2], ABDULATIF ALABDULATIF[3], (Member, IEEE), ABIODUN ESTHER OMOLARA[4], AND OLUDARE ISAAC ABIODUN[4]

[1]Department of Computer and Mathematics, Universiti Teknologi MARA (UiTM), Shah Alam 40450, Malaysia
[2]Department of Computer Science, Abu Dhabi University, Abu Dhabi, United Arab Emirates
[3]Department of Computer Science, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[4]Department of Computer Science, University of Abuja, Gwagwalada, Nigeria

Corresponding authors: Moatsum Alawida (moatsum.alawida@adu.ac.ae) and Olayiwola Tokunbo Taofeek (2021837596@student.uitm.edu.my)

This work was supported by the Office of Research & Sponsored Programs of Abu Dhabi University, United Arab Emirates.

ABSTRACT The exponential increase in the compromise of sensitive and intellectual properties alludes to the huge price the global community must pay for the digital revolution we are currently experiencing. This irrefutable reality is a major reason why cybersecurity defences continue to be a pressing and timely area of research. Traditional countermeasures of cyber defence using boundary controllers and filters such as intrusion detection, access controls, firewalls and so on, have proven ineffective. Such measures fail to account for the attacker's inherent advantage of being increasingly techno-savvy, as well as their persistence in attempting to compromise the security of not only high-value targets, but also the vast pool of oblivious users of technology. The use of decoys and deception is one of the emerging solutions for cyber defence. Leveraging decoys and deception for security pre-date the advent of the digital revolution as centuries have witnessed the military using human decoys to deceive and successfully defeat their adversaries during wars. However, its benefits for reducing cyberattacks in these digital times have not been thoroughly investigated. One of its use requires that fake text documents are positioned in the repository of critical documents in order to mislead and catch hackers attempting to exfiltrate sensitive documents. Current methods of generating fake text documents involve using symbols, junk documents, randomly generated texts. Such approaches fail to capture the empirical and linguistic properties of language, resulting in messages that do not scale well, are not realistic, fail in the context of syntax and are semantically void. Consequently, failing to convince the attackers to believe they are the original messages. This paper presents a Cognitive Deception Model (CDM) based on a neural model which takes an input message and generates syntactically cohesive and semantically coherent independent looking but plausible and convincing decoy messages to cognitively burden and deceive the adversaries. The experimental results used to validate the models, as well as the comparison with state-of-the-art tools, show that it outperforms existing systems.

INDEX TERMS Artificial advanced persistent threats (APTs), cyber-attacks, cyber defence, deception, decoys.

I. INTRODUCTION

The ubiquitous state of the world implies the inevitability of the utilization of computers, smart devices, and their related components in the execution of people's day to day tasks. This convergence of the physical and digital world implies that most valuable information is now in digital formats. Hence, the inception of cyber-attacks [1], [2]. Cybersecurity concerns have been on the rise due to the tremendous increase in the prevalence, number and complexity of cyberattacks [3]. The alarming number of reports where cyber attackers have undermined and compromised the confidentiality, integrity

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

and availability of digital information for remunerative and other malicious purposes remains worrisome and have put individuals, businesses, the government, research community, amongst others, into a state of unrest.

Social engineering, ransomware, phishing, and other sophisticated and orchestrated attacks continue to target and ravage people, industries, and government networks to exfiltrate sensitive data. According to the recent Verizon's 2021 Data Breach Investigation Reports (DBIR), organizations, regardless of size or industry, are tasty targets for cybercriminals and thus, they continue to remain at risk of cyber-attacks. Verizon reported over 29,207 incidents with 5,258 confirmed data breaches [4], [5]. Theft of people's personal credentials has resulted in personal e-mail compromise (PEC) causing massive financial losses. Furthermore, theft of business email accounts has resulted in business email compromise (BEC), in which cyber attackers compromise corporate email accounts and then deceive employees to perform specific tasks such as making a wire transfer to a fraudulent bank account or sharing confidential organizational details [6]–[8]. Employees who work from home may be vulnerable to several attacks, especially in this unprecedented time with the COVID-19 pandemic [4], [5].

Cyber-attacks are often thwarted through the use of security measures like encryption [9]–[12], access control [13], and others. One of the emerging methods for combating document exfiltration during cyberattacks makes use of deception-based mechanisms, which focus on exploiting adversaries' biases and altering their perception by supplying them with bogus or fake data to waste their time and resources. This strategy provides the attacker with a plausible alternative that appears to be the real or actual data.

The deception process involves the insertion of fake attack entry points, for instance, an open service port which an attacker may expect to see in a typical network system. The attacker will then scan through the network to connect to the open service, unaware that he has been misled to a fake server (commonly called honeypot). If the deception strategy is appropriately implemented, then the attacker will be led through a controlled process path that consists of broadly three attack stages namely; scanning, discovery, and exfiltration. The scanning stage is the process where the adversary is searching through available means of exploitable entry points. In the discovery stage, he discovers the entry point (which may be real or fake). At the exfiltration stage, he will start exploiting and stealing intellectual properties. A well-constructed deception system must handle the exfiltration stage well by generating deceptive content that looks real to keep the attacker busy while wasting his time, resources, and cognitive effort. The stages run in tandem with an exposing stage where the activities of the attacker are already under observation. The network administrator then takes the response actions.

This paper concentrates on the exfiltration phase where content-based cyber defence solutions are applied to text documents to generate fake documents known as decoy files, honey files or fake files. However, generating fake content that produces plausible and believable text data which can successfully deceive attackers is a challenging task.

The current generation process of fake files yields random characters, symbols, texts and often gibberish which fails to convince the adversaries to accept the decoy data as the real data, especially in the practical scenario when applied in real-world use-cases. Hence, limiting the use of deception-based cyber defence systems and leading the adversaries to continue to traverse the system until they achieve their malicious aims. Moreover, recent work that has attempted to improve on plausibility and believability of the decoy messages focuses on the textual characteristics of the sentences in the document and does not consider the correctness and completeness of the messages based on the quality of its knowledge [14].

This paper presents a cognitive deception model (CDM) for successfully deceiving attackers during the exfiltration phase of an attack. CDM is designed based on the neural language model and it takes the input (original) file to generate an independent looking but plausible and convincing decoy message to deceive the adversaries. CDM focuses on generating decoy files that not only focus on the textual characteristics but also on the completeness and correctness of the messages. The generated decoy message embodies the critical criteria of deception systems as it repackages, masks and mimics the real data while hiding vital information in the real document. Additionally, it has the advantage of generating decoy messages sharing a similar domain as the input file. Thus, convincing even a cyber-savvy attacker with domain knowledge of the real data. Further details regarding the proposed model's development, analysis, and implementation are substantiated in subsequent sections.

The rest of this paper is organized as follows: the background and related works on the current measures of tackling data exfiltration is presented in Section II. In Section III, preliminarieis to understand the flow of the methodology and the justification for using it is given. In section IV, the methodology is presented. Experimental results and analysis are presented in Section V. Section VI concludes the research.

II. BACKGROUND AND RELATED WORKS

Despite traditional approaches such as the use of access control, intrusion detection, malware scanners, firewalls and other prevention technologies have been used in diverse scenarios to curtail document exfiltration attacks on the cyber scene. However, the cyber scene still suffers from numerous attacks, as current traditional approaches have been insufficient at preventing network penetration which subsequently leads to exfiltration and theft of confidential documents and resources [15].

Deception-based techniques provide essential features and critical preferences over traditional security controls. The cyber deception technique leverages diverse strategies to control and mislead the attackers into taking specific actions (or inactions) during their attack. Such techniques use

entities such as decoys to perform the deception. A considerable amount of literature has been published on the use of deception. Several sectors have seen the use of deception to protect confidential resources. Decoy passwords, also known as honeywords, is used to detect the breach of passwords or a password vault/database [16].

HoneyDetail was suggested for ensuring patient's information privacy and thwarting electronic health record threats based on decoys in the medical domain [17]. Deceptiver was proposed as a centralized server that can be hooked to public servers as real production public-facing servers to deceive adversaries by injecting decoys to alter their responses [18]. HoneyFiles, honey documents are files or documents presented to attackers as original files or documents to mislead them during data exfiltration [19].

HoneyFiles and honey documents are constituents of a complicated/complex distribution (for instance, e-mail messages). Thus, their decoy construction requires the generation of fake but syntactically, semantically and contextually realistic natural language to convince and successfully mislead an attacker into taking or not taking action during file exfiltration [20]. The complexity of such decoy production has led to researchers taking various approaches to handle the problem. For instance, Salem and Stolfo used a strategy of modelling user search behaviour. The effectiveness of their approach relies on the attacker's behaviour of searching for documents differently than authentic users. Their approach does not consider making the contents of the document to be realistic. However, the names and directory placements are real [21].

Other forms of production of the text contents for fake document generation are categorized into the use of random symbols and characters, the use of random words and sentences extracted from a specified public document or corpus, generation based on rules and preset template, generation from one language to another and recently, generation based on tools and techniques from natural language [22]–[24].

The advance in natural language, a subdivision of artificial intelligence, has brought about promising discoveries in helping machines disambiguate semantics in documents and mimicking language the way humans understand and use it.

Recent attempts to explore new strategies to develop decoy messages that are believable was given by Karuna *et al.* [25]. They presented a comprehensibility manipulation framework (CMF) which takes an original document as the input, identify and delete salient information. CMF generates fake documents that are hard to comprehend, contain no significant information but are readable and believable to the attacker. They additionally proposed another strategy of generating fake documents by manipulating text comprehensibility for cyber deception [13]. While their approach produces promising results that yield believable and interactive fake documents, it captures mostly the syntactic and little of the semantics of language. Their approach depends on a semantic similarity model to provide the similarity of noun phrases only without considering other phrase types and clausal levels. Such construction produces decoys that may fail to convince a language-savvy attacker with knowledge of the language used to represent the document.

Additionally, their approach and other current approaches of using random texts, symbols and characters do not account for completeness and correctness, which are attributes that shows the quality of knowledge from the generated document and which plays a vital role in showing the plausibility of the content of the documents to aid in deceiving the attacker.

Another challenge with the current file generation strategy is adaptability, especially where the decoy files are not adapted to different domains and as such, a cyber-attacker may learn partial information from the decoy files during his attack with a high interaction system. The attacker may also maul the network system to learn a part of the content of the message in cases where some messages are substituted. Therefore, the goal of this research is to address the shortcomings and gaps highlighted as challenges in the current honey file generation method.

A. NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) aims to allow computers intelligently process human languages. Researchers in the field has categorized the advancement made in NLP systems into three waves; the first wave was based on a rationalist approach which is dependent on the design of handcrafted rules incorporated into NLP systems based on the assumption that knowledge of language in the human mind is fixed by genetic inheritance. The second wave, been an empiricist approach, is dependent on the consensus that rich sensory input and the observable language data in surface form are required and sufficient to enable the mind to learn the detailed structure of natural language. This prompted the development of probabilistic models for discovering the regularities of languages from large corpora. The third wave is base on a deep learning approach which exploits hierarchical models of nonlinear processing, inspired by biological neural systems to learn intrinsic representations from language data, in ways that aim to simulate human cognitive abilities. The intersection of deep learning and NLP has resulted in striking successes in real-world tasks [26], [27].

The neural language model (NLM) specifies the representation that takes the meaning of the input text into account. They capture the rich semantic properties of language and are more appropriate for understanding language. Recent work in natural language are majorly focused on using the neural language models to get better results as they are promising sources of capturing human cognitive capabilities [28], [29]. The dominating neural models in use now is the Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN).

The RNN was first proposed by Mikolov *et al.* It has a linear recursive structure which is a desirable property for capturing and processing sequences of data [30]. The RNN produces an output and a state by taking as input, an element of the input sequence at each phase. In the next

phase, the state is used to condition the next output and it denotes the memory of the prior input and output. An RNN cell determines the way in which the input and previous state regulate its output and next states. RNN enables the encoding of dependencies between the inputs. Nevertheless, when used on long data sequences, it may cause an explosion and varnishing state against its gradient. The standard LSTM upgrades the RNN by processing sequences in a temporal order while ignoring past contexts. The meaning of a specific word in a sentence is dependent on future words and not just previous words. Thus, the bidirectional LSTM (BiRNN) was introduced as an extension of the standard LSTM The BiRNN is powered with two cells, one of the cells is used for the original sequence order for forward states and the sequence order for the backward state is from the second cell. Basically, it has a second layer where hidden-to-hidden connections flow in opposite temporal order, thus allowing information to be explored from the past and future.

CNN is a neural network with one or more convolutional layers. The layers compute the convolution of sections of the input. These sections are defined by size, window, and stride The convolutions are weighted, and a bias is added to train the results. The convolution, bias and the weights all combine to form a filter.

Collectively, the RNNs are biased with respect to the last item in the sequence because of their forward sequential nature. However, their recursive capability is suitable for language modelling as they allow processing of sequences of tokens in a text while keeping track of the state that represents the memory of the previous tokens. The CNN does not have the bias. The bias are added during the model training. However, they are unsuitable for learning long distance semantic information, especially for systems having varying length of inputs, as their size of filters is dependent on the length of the input sentence. While padding of the input sentence can be used to adjust the length, they skew the results when the padded tokens are used in the convolutions.

Other NLM such as the attention-based neural network (ABNN) pays attention to specific word. The basic concept driving the attention based neural network is that there might be relevant information in every word in a sentence. Thus, for the decoding to be precise, it needs to take into account every word of the input, using attention [31]. While the ABNN is useful, it adds more weight parameters to the model, thus, increasing processing especially in human-generated text where the input data for the model are long sequences. Variants of ABNN such as BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-Training) by Open AI, and newer versions such as GPT-2, RoBERT, ESIM+GloVe and now GPT-3 are based on the attention and usually referred to as Transformers [32], [33]. Transformers process an input sequence of words all at once, mapping relevant dependencies between words no matter how far apart they appear in the text. As a consequence, they are highly parallelizable, could also train much larger models at a fast pace, and can use contextual clues to solve numerous ambiguity issues in text. However, the transformer-based NLM are yet to be fully validated. According to a recent Google Research study, some of the hype in the capabilities of the transformer-based NLM and the improvement made on other of its variants did not improve its performance. Thus, limiting it widespread adoption in its realization in some real-world use-cases. Thus, the justification of leveraging RNN for this study. RNN are more stable as they have been used for a while [34], [35].

III. PRELIMINARIES

This section discusses some components required for understanding the underlying theories guiding the methodology that is used in developing the proposed model.

A. SEMANTIC AND SYNTACTIC RELATIONS

The semantics of the language in a document is concerned with the meaning assigned to the symbols, characters, or words. In contrast, syntax is concerned with the structure or grammatical form of a language. Coherence in the semantics and cohesion in the syntactic relations between words are needed to produce plausible, realistic but decoy sentences from the document. The semantic coherence is constructed to reflect a logical and plausible sentence that makes sense as an entity by dealing with the meaning and interpretation of the words. The grammatical structure of the sentence is enveloped by cohesing the sentences in the document syntactically.

The context in which a word is used is vital and must be taken into consideration as a specific word may have more than one semantic meaning. Different relations among distinct categories are used to show this relation such as antonym, synonym, hypernym. Relating this context to our long-term goal of introducing plausibility to form realistic messages, a dictionary of English words that will do the mapping to create the wordlist for the decoy file is required. Our proposed model adopts the WordNet dictionary for this purpose. Several studies have shown how WordNet outperforms other dictionaries as it takes into account how humans process natural language [36]–[38].

WordNet contains an extensive network of 155,327 words. It is one of the largest thesauri and ontology repositories for word meaning. It contains 207,016 different word senses. These word senses belong to 117,597 different synsets and are ordered as a synonym set called synset. Each synset consists of a list of synonymous words that denote the same concept and are interchangeable in many contexts grouped into unordered sets. A synset can also be seen as a grouping of words that share a common meaning. The synsets are interlinked by means of conceptual-semantic and lexical relations. Thus, making it a useful tool for natural language processing and computational linguistics. Basically, synsets are connected to other synsets by means of semantic relations.

Semantic relations hold among all members of the linked synsets. For instance, a noun part of speech word U is a

hypernym of V if every V is a (kind of) U (canine is a hypernym of cat). Synsets contain a textual description of the semantic meaning of the synset. A word could have many possible senses (synsets). It can also be understood as WordNet storing terms in synsets, and every synset having relations to other synsets. The relations can be a synonym, hypernym, meronym or hyponym. All synsets have an id that can be used to uniquely identify them, and it is possible to extract all synsets meeting certain requirements from Wordnet.

The proposed model is constructed by using Semantic relations that describe relationships built among the synsets. Also, lexical words with similar meanings are linked using semantic-related pointers.

IV. PROPOSED COGNITIVE DECEPTION MODEL (CDM)

This section concentrates on the development of the Cognitive Deception Model (CDM). The CDM is a convoluted representation which converges several fields to address the major problem of cyber deception by cognitively burdening and confounding the activities of the attacker. To address the challenge of the quality of knowledge of the sentences in the fake document that will be generated, plausibility must be introduced into the CDM. The plausibility will be responsible for capturing semantics and syntactic relations to form contextually realistic sentences that scales. Additionally, to introduce completeness to the decoy and maintain the secrecy of the real document so as not to benefit an attacker that may try to decipher the message, a domain/field-specific enhancement will be made in the CDM.

A. CONSTRUCTION OF THE CDM

The CDM is constructed starting from addressing the plausibility of the sentences in the documents. Sentences are processed after punctuation and treated each as a new line.

The plausibility characteristic is introduced by leveraging a combination of techniques and ideas collected from the field of cryptography, natural language processing and linguistic standpoint. The CDM reads each word in the sentence and disambiguates the underlying context by understanding its structural constituents in terms of meaning and transforms it as a syntactically and contextually realistic sentence by preserving the ontology with respect to the part of speech (POS) each word in the original document belong to. This approach takes its perspectives from the conceptual designs in the work of [10], where messages are translated to semantically equivalent messages.

However, the proposed CDM is different as it leverages a number of scrambling techniques of transforming or otherwise re-encoding words with other words. It also reorders the sentence. This concept was adopted from the diffusion and confusion principles used in cryptography. In cryptography, confusion makes the relationship between the key and the ciphertext as complex as possible (the same approach applies to the relationship between the ciphertext and the plaintext). For instance, in the Caesar Cipher, a letter in the alphabet is re-encoded by using another letter three (3) positions down the alphabet to replace it and so forth.

Diffusion hides the relationship between the plaintext and the ciphertext by spreading the plaintext characteristics over the ciphertext. The confusion process of the proposed CDM uses words to represent other words that share the same part of speech. The diffusion process manages to keep the same syntactical structural characteristics of the sentences in the main document into the decoy message. However, in some cases, the syntactical structure is different due to the scrambling process during the production of the decoy message.

B. DEVELOPMENT OF THE CDM

The CDM is developed using two steps.

In the first step, the dataset is prepared, preprocessed and a relationship function is built. The dataset is defined in this context as the representative document which is processed line after line after punctuation.

In the second step, a work in process (WIP) decoy message is processed for classifying and intelligently smoothing the WIP decoy message to produce the final product which will be an independent looking but plausible and convincing decoy message to deceive the adversaries.

C. DATA PREPARATION AND PRE-PROCESSING

The data preparation method is important to get the most out of the data to get a good predictive model for the decoy message which will be generated from the texts in the main document. Additionally, the data must be prepared such that data leakage will be avoided which may lead to a poor model. Messy data, which may be missing values, outliers, etc would be identified and handled during the production process. The development and implementation of the models require that the data are converted to numeric vectors based on some specific requirements.

Sentences are split into smaller units called tokens. The tokens are depicted numerically by converting them into a sequence of numbers for the computer to calculate with them. Words are represented as either continuous or discrete entities which are a representation of binary vector that is all 0 values except the index of the word in the vocabulary, which is marked with a 1. Prior to tokenization, the text is preprocessed by changing the textual content to lower case and any format annotation, such as HTML tags are removed. The preprocessing stage incorporates stop word removal and segmentation. Stop word comprises of English words that are mostly used but do not provide any concrete contribution to the semantics of textual content. Examples of such classes of words are determiners such as a, the and part-of-speech words belonging to the category of pronouns, prepositions, etc. Removal of such words improves accuracy, time and saves memory space by increasing the speed of processing. Segmentation divides the texts into meaningful units of words or group of words as there are certain groups of words that stand as an entity and cannot be split into a unit without causing a syntactic failure.

Thereafter, the characters are divided into various categories of units such as symbols, words and sentences. Tokenizing characters means splitting the sentence into smaller units of symbols. Tokenizing words means splitting the sentence into smaller units of words. Tokenizing sentences means splitting the sentence into smaller units of sentences, in some cases, phrasal or clausal forms.

The output of tokenization is thereafter converted to a data frame for better text understanding and for further processing such as Part-of-Speech Tagging. Depending on how the plaintext is processed, in some cases, the stop words are not removed if they form compound expressions or multiword expressions such as phrasal verbs, prepositional phrases and are important in the context used. Additionally, subject-verb and object relationships are considered to ensure that subject and verb quantities agree.

A grammatical analysis is done to parse out the features of each word. The grammatical analysis also encompasses analysis to determine the subject, object, verbs- such as-active verb, passive verbs. This is followed by a POS tag where each word is specified based on the part-of-speech it belongs to. The plaintext is then reconstructed with the dictionary of the wordlists from WordNet by encoding the string based on the classes. The POS is used to determine the classes from the WordNet to use as input. In the next section, the encode relationship function is developed where each word is encoded (transformed) based on the identified synset group.

D. RELATIONSHIP FUNCTION

The POS in the English language consists of Nouns, Pronouns, Verbs, Adverbs, Adjectives, Interjections, Prepositions, Conjunctions, and Article. Word types are illustrated following different grammatical rules. WordNet makes the distinction between four of the essential POS in the English language, which incorporate nouns, verbs, adjectives, and adverbs. The aforementioned are constituents of words that carry important meaning [38]. Nouns are classified as to entities, concepts, qualities, actions, states and can serve as the subject of a verb. The words that are categorized into verbs may serve as the predicate of a sentence to describe a state, occurrence, or action of existence. Adjectives belong to the classes that can modify the nouns. The adverb word is similar to adjective as they contain words which modify other POS words other than nouns.

Each word is processed based on their part of speech, the strings are appended based on their synset group until the complete string of sentences has been processed and replaced. The current state of the message is more or less a representation of their syntactical form which may have no meaning when looked at contextually. A parsing algorithm is used to deconstruct the sentence to find relationships and organize the words to establish subject-verb-object connections. For instance, relationships such as adjectives used with nouns, adverbs used with verbs, nouns used with verbs.

The relationship function is built by replacing the POS words extracted from the input document with its matching replacement in the WordNet thesauri. Each constituent is then used to build a tree that forms the WIP decoy message. The process of sentence formation is translated into algorithms, where each word is traced and updated in the tree using a function. The function to group the sentence is shown below:

Algorithm 1 Sentence Formation Function for WIP Decoy Document

Input: data (POS words)
Output: realistic words

```
function def updateTrace(trace,base):
    for i in range(len(trace)−1, −1, −1):
        if trace[i][0] == base:
            return trace[:i+1]
        clause = trace[i][1].findChild(base)
        if clause is not None:
            trace = trace[:i+1]
            trace.append([base,clause])
            return trace
    return None
def updateTree(trace):
    for i in range(len(trace)−2, −1, −1):
trace[i][1].updateValue(trace[i+1][0],trace[i+1][1])
            return trace
```

This stage produces realistic messages that scales. It accounts for completeness and correctness which are attributes that show the quality of knowledge from the generated document and which plays a vital role in showing the plausibility of the content of the documents to aid in deceiving the attacker. For instance, the input message,

→ the tiger is eating a fox

may produce a fake message such as,

→ the cat eat a mice

It will not produce a fake message like ''cat eat biscuit''. This is because the cat has been traced to be a carnivore and so the verb 'eat' will relate to carnivoral eating. It is also possible for the verb 'eat' to have been replaced with 'jump' and the nouns 'cat and mice' to have been replaced with 'dog and bird', resulting in another sentence altogether but which has meaning and is not semantically void.

Depending on the sentence, it is possible that the complete sentence may have changed due to certain words belonging to several synsets. For instance, when a noun is also a verb and another synset verb is used to replace it. An example is a word 'shot' which has two senses in the synset connecting shot as a snapshot or shot as an injection. This can generate fake messages such as,

→ the cameraman took a shot of me

or

→ the physician gave me a shot for my flu

the former relating to snapshot and the latter relating to injection. The proposed approach benefits from such decoy

TABLE 1. Shows the processes of the syntactic and semantic processing.

Design Process	Details
Segmentation	The process divides the plaintext into distinct units.
Stop-word Removal	The process removes stop-word which comprises of English words that are mostly used but does not provide any concrete contribution to the the semantics of the plaintext.
Tokenize Word	The plaintext is split into smaller units called tokens. The tokens are described in a numerical way by converting them into a sequence of numbers for the computer to calculate with them.
Lemmatization	Removes the text's ambiguity by reducing the word's density from the given plaintext by converting all words having the same meaning to their different representation in their base form.
Morphological Segmentation	Process divide words into individual units.
Part-of-speech (POS) Tagging	The process identifies the part of speech for every word.
Parsing	The process undertakes grammatical analysis for the provided sentence.
Sentence Breaking	The process places sentence boundaries on the plaintext.
Stemming	It involves cutting the inflected words to their root form.
Word Sense Disambiguation	The process determines the parts of the text that can be identified and categorized into preset groups. Examples of such groups include names of people and names of places.
Named Entity Recognition	The process gives meaning to each word based on the context.
Natural Language Generation	Generate natural language message

messages due to each synset having one or more lemmas, which constitute a particular sense of a distinct word.

E. CLASSIFICATION

The classification stage is an extra function attached to the CDM to make the end product of the decoy message to be field-specific; that is, its adaptability to different domains. This stage is important to mislead a cyber-savvy attacker with domain knowledge of the real data. The classification used in this study is based on deep learning. Deep learning was adopted because it allows the achievement of state-of-the-art results on difficult problems such as understanding context and knowledge in large datasets, object recognition in photographs, amongst others.

A recurrent neural network (RNN) that incorporates a bidirectional long short term memory (LSTM) is used for classifying the WIP decoy message generated from the relationship function in the previous section. This paper adopted a similar approach used by the work of [1].

A major advantage of BERT is that it generates "contextualized" word embeddings/vectors, but this is also its biggest disadvantage because it is computationally intensive at inference time, implying it can become very expensive if it is to be used in production. The deception system is a highly interactive production system and needs to be deployed in an effective and cheaper mode, thus the BERT system and others such as GPT may be unsuitable for it use for creating a deception model. Additionally, RNN has been used for decades and it is more stable for creating synthetic text and its performance can be validated.

The bidirectional RNN is consolidated using a memory gating mechanism, the LSTM. The LSTM model is used to classify the domain to which the input message falls into before generating a decoy message falling into similar domains. A hidden Markov model (HMM) is then used for generating language as it is efficient in listing elements from a family of strings. It has a finite internal state which can generate a set of observations referred to as external events that can hide the internal state changes to a viewer outside the system. Precisely, it scans through the identified domain and then generates decoy messages sharing similar domains as the WIP decoy message.

The RNN understands text data as a signal consisting of words. It consists of an input layer, a hidden layer and an output layer. The sentences are processed in multiple layers that allow information to be persistent. The input layer at time $\not{t}$ along with the hidden layer at time $\not{t}$ are assembled as a new input layer to compute the hidden layer at time $\not{t}$.

In the input layer, the trained datasets which are sentences with labels are pre-processed. The context data are extracted from diverse domains in public texts, dictionary, Wikipedia and others. Pre-processing is done to remove redundant data such as spaces, special characters. Letters, numbers, tenses and special expressions in English are not removed. Sentences are converted into sequences of word indices $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3, \ldots, \mathcal{W}_n$, implying each word have an index which is the integer ID for that specific word.

The steps to pre-process the data is given algorithm 2:

The sequence of text which have been tokenized into words $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3, \ldots, \mathcal{W}_n$ are passed to the embedded layer of the LSTM. The sklearn module of the TensorFlow library is used to generate and produce the word vectors of the text, which is then sent to the embedding layer as features.

In the hidden layer, the LSTM augments the RNN node with the LSTM cell, which is devised to save the text history information. The LSTM leverages three gates (input gates, forget gates and output gates) to control the usage and update the text history information. The gates decide when the input

Algorithm 2 Steps to Preprocess Data

Input: realistic words from WIP ((In Algorithm 1)
Output: realistic words with connected with correct context
1: Decompose WIP decoy message into domain label.
2: Prepare data by removing overly common words WIP decoy message (Stop word removal).
3: Determine the maximum length of the WIP decoy message.
4: Tokenize the WIP decoy message to words (Tokenization).
5: Words are encoded into numbers
6: Determine the vocabulary size from the tokenized words.
7: Sequences are padded to the maximum length and processed using one-hot encoding.
8: Split the data into train and test.
9: Data is then passed to the HMM.

to the LSTM is sufficient enough to remember when it should continue to recollect or forget the value, and when it should yield the output of the value. Basically, the memory cell in the LSTM and the three (input, forget and output) gates are equipped to allow the LSTM to read, save and update long-distance history information of the data. The LSTM uses four main steps to carry out its operation in the gates.

The steps are given in algorithm 3:

Algorithm 3 Steps to Classify WIP Decoy Message Into Domain

Input: realistic words from WIP ((In Algorithm 2)
Output: domain-connected realistic words
1: Compute the values of the input gate and the forget gate
2: Update the steps of the LSTM cell
3: Compute the value of the output gate.
4: Update the output of the whole cell.

The importance of the first step is for the forget gates to determine which information from the cell state to be discarded at that stage. The second step is required for the input gate to decide which new information is to be stored in the cell state. The importance of the third step is to determine the updated value of the cell state based on the last input gates and forget gates information updated. The fourth step determines the value of the output based on the state of the cell. This computation is done using standard sigmoid and tanh functions which are omitted to remove redundancy but can be found in [39], [40].

The gates make it computations based on selected features, for instance, ''My name is Lula. I love children. I have a daughter born in the year 2013...<more conversation>... In 2022, I will have another child. In early 2023, you could say I have how many number of children...''. Handling information like this requires the cell to process which information it considers to be important and which is not. For instance, the third sentence ''I have a daughter born in the year 2013'' must be kept to be able to predict the last sentence ''In early 2023, you could say I have how many number of children...'' accurately.

Word embeddings are used to map semantic meaning into a geometric space by associating a numeric vector to each word in a way that the distance between any two word vectors encapsulates a significant proportion of the semantic connections between the associated words. The two words, 'maize' and 'tiger' are far different semantically and so an embedding space would represent them as vectors that are far apart. The words, 'food' and 'kitchen' will have close embedding space as they are related,

→ food can be cooked in the kitchen
→ she ate the food in the kitchen

The embedding space is the geometric space formed by the vectors. The embedding layer generates the word embeddings by multiplying an index vector with a word embedding matrix. The sentences were padded in the pre-processing stage so that the input to the model can be of about the same size with the decoy message that will be produced as the RNN model needs a vocabulary size and the maximum length of the sentence. An embedding matrix which will contain at index n, the embedding vector for each word of index n is developed in the hidden layer.

A fully-connected layer applies a non-linear function to the concatenation of word embeddings of n previous words. The connected layer yields an intermediate representation of the input. The ReLu function was used for the linear activation function in our CDM development.

In the output layer, sentences are then passed line-by-line with the label which is generated as vectors of different domains, d1, d2, d3,..., d m. In summary, the embedding layer LSTM (a,b,...,m) map the integer inputs to the word vectors $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3, \ldots, \mathcal{W}_n$ found at the corresponding index in the embedding matrix to output the different domains, d1, d2, d3,..., d m. The features are extracted from the text by the embedding layer and sent to the fully connected layer for classification. Softmax classifier is used for developing the CDM. The softmax classier maps the output of neurons to (0,1) interval and picks the class of text with the highest likelihood value to output.

In summary, the BiLSTM-RNN model processes the sequence of input vectors and are densely connected, allowing features of each word vectors to be extracted as they are computed in each layer. The LSTM helps to classify the input message to which domain the underlying plaintext falls into. Then based on the classified domain, the HMM model would look into only the identified domain data and then generate a new sentence.

F. DETAILED DESCRIPTION OF METHODOLOGY

As described in previous sub-sections, each word is processed based on its part of speech. The strings are appended based on their synset group until the complete sentence string has been processed and replaced. The state of the message at WIP is more or less a representation of their syntactical form, which

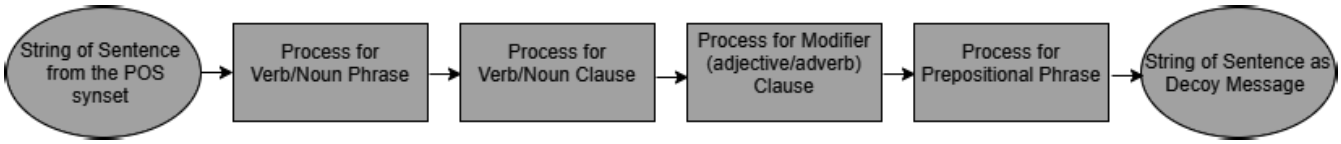

FIGURE 1. Process sentences for decoy generation.

may have no meaning when looked at contextually. A parsing algorithm deconstructs the sentence to find relationships and organizes the words to establish subject-verb-object connections. For instance, relationships include adjectives used with nouns, adverbs used with verbs, nouns used with verbs.

Recall that the Relationship Function in subsection (IIIe) generates the POS word synsets, and the output is appended to form the string of sentences. The string of sentence is further processed based on its phrasal form or clausal form. This grammatical analysis stage establishes the subject-verb-object relationship to ensure the decoy messages generated form meaningful sentences.

It also helps to restructure the sentence. This is important to capture syntax while enforcing meaning in the sentences generated. To put it in context, word level processing has been done and there is a need to cater for intensifiers and modifiers which forms a phrasal or clausal relationship. The phrasal or clausal connections leads to the sentence, in this case, the decoy message.

Another reason is to control the production of semantically void sentence, where syntactically it may be correct but meaning-wise, it is empty. The parsing algorithm uses a scrambling method to deconstruct the sentences to enforce the relationship based on the Stanford dependency parser. A high-level diagram showing the process of generating the decoy message is given in Fig. 1.

Supervised The approach adopted for this research helps to hide the length of the sentences/plaintext message. It attempts to keep the decoy message as compact as possible while restructuring the sentence. To put it loosely, it forces the output text sequence of words to be syntactically and semantically correct from a linguistic standpoint. While we have no control of the length of the message, the matching of POS words in the word-level, aside catching grammaticality also helps to maintain the conciseness of the decoy message as much as possible. The breakdown of each process is given using the flowchart depicted in Fig. 2 for encoding and Fig. 3 for decoding.

The flowchart in Fig. 2 is for encoding and while Fig. 3 for decoding. In the encoding phase, the generated sentence from the POS tag from WordNet is processed for noun phrase to check if a group of word in the sentence is functioning as the subject or object. After establishing if it is a subject or object in the sentence, then it will be used to form the first string or appended to an already growing string of the message. The same algorithm will be run for verb phrases to establish subject/object connections. The algorithm is used for the prepositional phrases in the sentences. The same algorithm is run for intensifiers and modifiers; the adjectival and adverbial clause until all the sentences have been processed. The sentence is then decoded to its word representation as shown in Fig. 3.

To translate the process of sentence formation into algorithms, each word is traced and updated in the tree using the function in algorithm one described in Section IIIe.

V. EXPERIMENTAL RESULTS AND ANALYSIS

Another The evaluation of decoy documents from real/original documents requires a human level of understanding. Language generation in open domains requires human comprehension and background knowledge [41]. Qualities such as the accuracy of the semantics and syntactic content of the text document can be used as attributes for deducing the correctness of a message. Cyber attackers are humans who can decipher if the message is convincing or believable. For instance, a distinction between grammatical and ungrammatical sentences can quickly be noticed by a native speaker of a particular language. He/she can infer knowledge based on the perceived correctness of semantics and syntactic and also based on completeness. Therefore, a test to check the believability level based on a decoy turing test (DTT) is adopted for the evaluation of CDM.

The DTT test was introduced by a Mathematician called Alan Turing who invented the theory of digital computation. He programmed a computer to simulate human thought processes using an imitation game [42]. The imitation game determines the level of artificial intelligence of a machine through the inability of a human judge to distinguish between a human acting as a human conversational simulator and a machine by engaging both in a conversation.

Based on the imitation game approach, the proposed CDM is evaluated using human judgement considering that the believability of the decoy document during the exfiltration is carried out by humans and the human subject will be the one to accept or discard the decoy message during the attack. Thus, this experiment will test if the attacker can detect if the decoy message is the real message or not. Details of evaluation are given in the following subsections:

A. DATA COLLECTION AND MODEL PREPARATION CHALLENGES

The survey was done by human subjects/judges consisting of 25 volunteers. We used 25 volunteers due to the practical limitations of the user study. The volunteers were selected

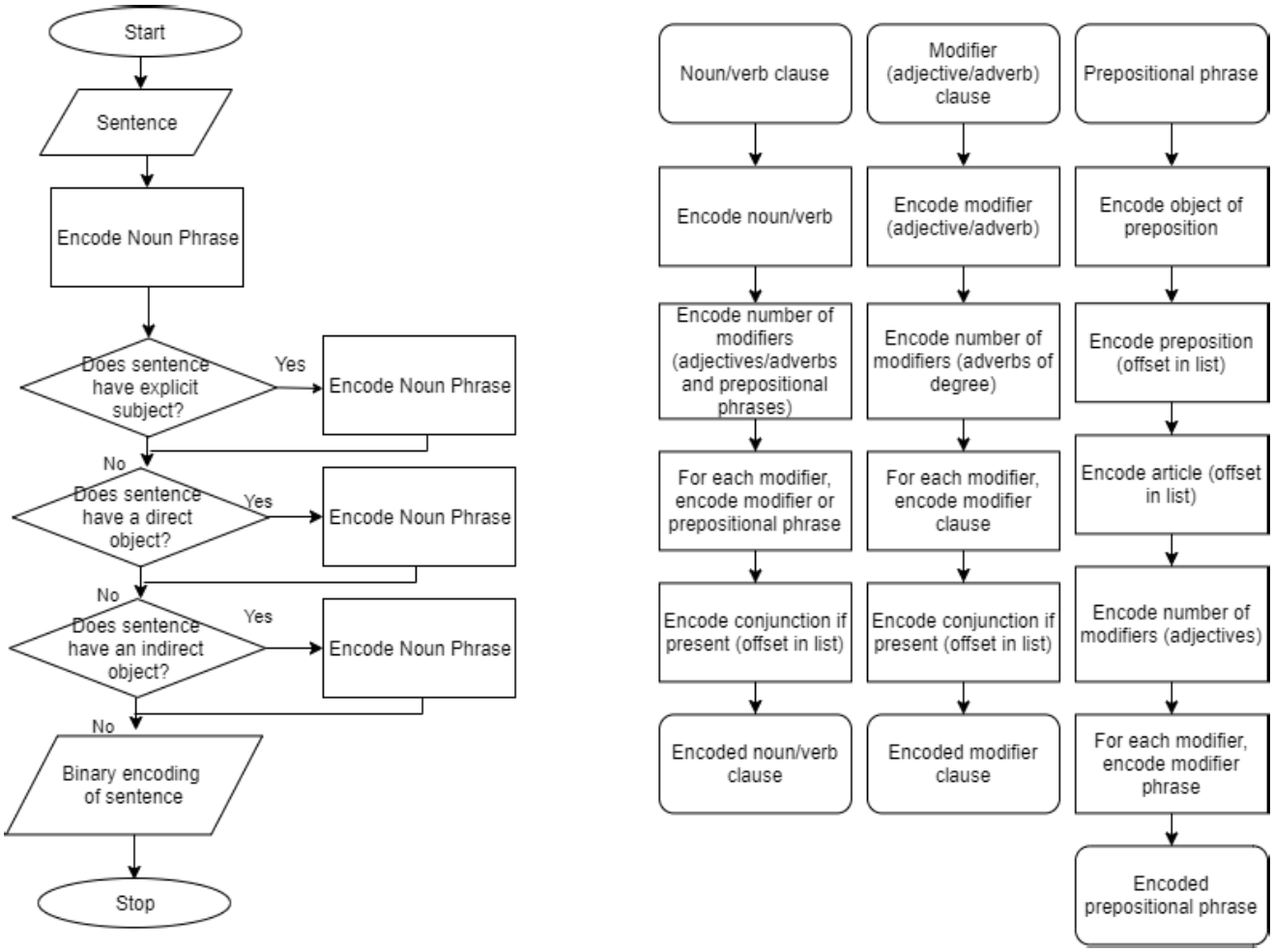

FIGURE 2. Flowchart for encoding a message.

based on their mastery in English language and computer security. Representative text for the original document was extracted as sentences from standard English corpora. Additionally, three selected domains were used, namely; keywords from movies, weather and restaurants. We considered using these three domains because our research tentacles restrict us from reaching experts in other domains. The selected three domains are not so in-depth and so most people can easily point out any keywords from such domains. Thus we used the same 25 human subject volunteers to evaluate the domain-specificity of the CDM model as well.

Entities from the domains were extracted from their specific corpus on the Internet. Statistical metrics was used to extract the domains by selecting the keywords that occur more frequently in the selected domains. The Kullback Leibler divergence was used to identify the words that are specific for the selected domain by differentiating it between a general corpus and a domain-specific corpus. A similar approach was used by [41]. The domain-specific keywords are fed to the CDM model to train it to effectively learn and create new sentences from such domains when an input sentence is given.

However, the proposed CDM model is generalizable for any domain as long as some keywords for the needed domains are fed to the model. Sentences are processed line by line after punctuation. Each document contains about 750 words of ten sentences each so as not to burden the judges with lengthy texts. A ssummary of the instrument used to collate the human judges is presented in Appendix A.

B. EXPERIMENT

The representative texts are fed as input into the CDM model to generate the decoy document. For each sentence, the input and output sentence is passed to the human judge to distinguish which of the document he/she perceives as the fake document. Table 2 presents the results of the simulation. Also, Table 3 presents the sample dataset from the input sentences and output sentences in the document.

Based on Table 1.1 given above, 0.5% which represents two of the 25 volunteers were able to detect the decoy document when it was based on English texts. Additionally, none of the 25 human judges where able to detect the decoy documents when it was domain-specific. The experimental

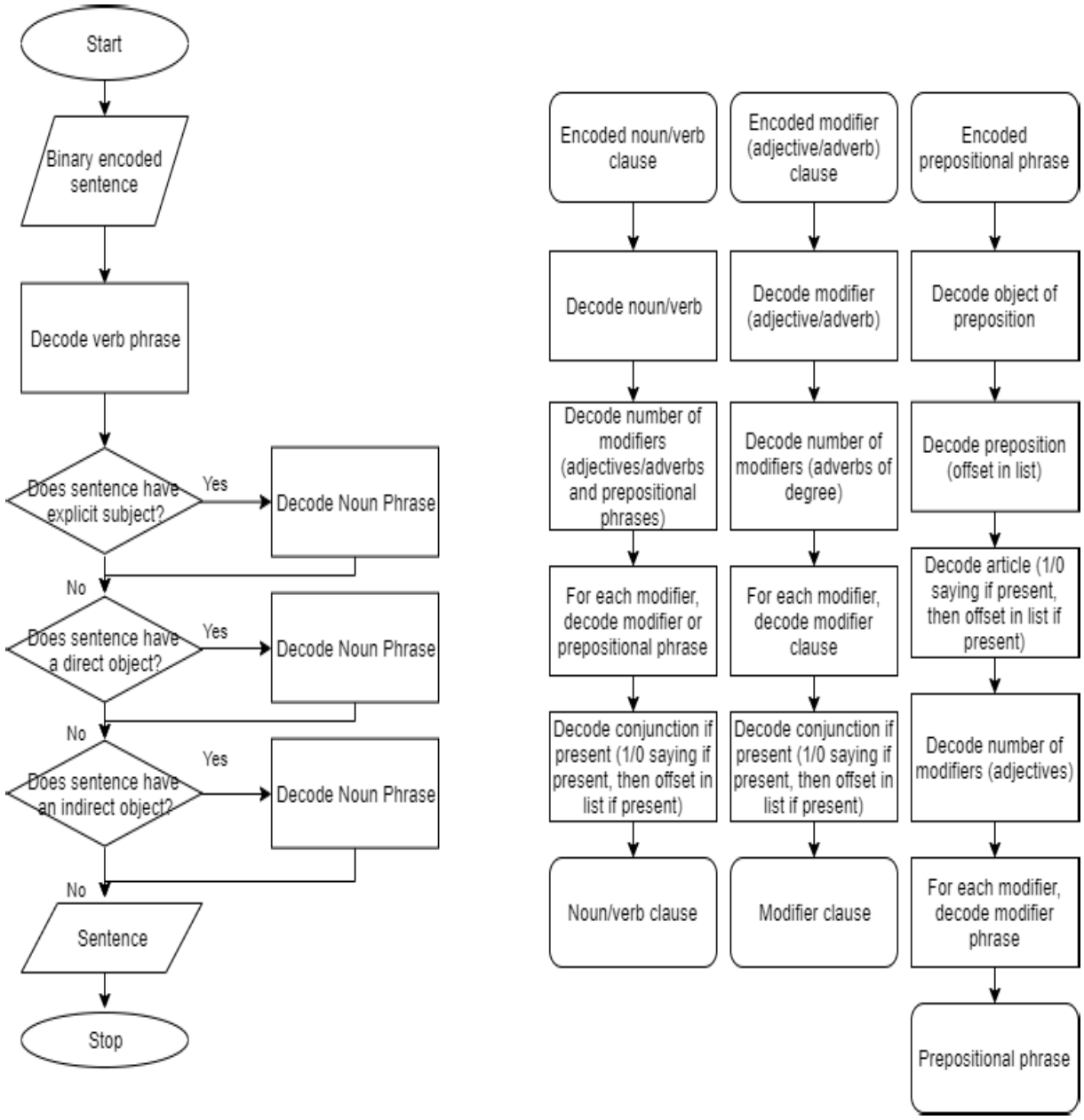

FIGURE 3. Flowchart for decoding a message.

results show that the likelihood of distinguishing between the original document and fake document is significantly low.

Table 1.2 shows us some of the input sentences and output sentences from the original input document and the decoy output. The similarity between the datasets from the input and output documents makes it difficult for an attacker to make a distinction and thus he can be misled into believing the decoy document for the original document.

C. STATISTICAL EVALUATION BASED ON LEVENSHTEIN DISTANCE (LD)

The Levenshtein distance (LD) quantifies the minimum number of insertions, substitutions, or deletions required to transform the input (original) document into the fake document or vice versa.

The LD between two strings: string p and string q of length |p| and |q| is given by levp,q (|p|, |q|) where

$$lev_{p,q}(a,b) = \begin{cases} \max(a,b) \\ \min \begin{cases} lev_{p,q}(a-1,b)+1 \\ lev_{p,q}(a,b-1)+1 \\ lev_{p,q}(a-1,b-1)+1_{\left(p_{q\neq Q_b}\right)} \end{cases} \end{cases}. \quad (1)$$

if min (a,b) = 0

TABLE 2. Results from the simulation.

Simulators	Success Rate for Detecting Sentences in the Decoy Document based on English Language Words	Success Rate for Detecting Sentences in the Decoy Document based on the Three Selected Domains
1	0	0
2	0	0
3	1	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	1	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
Percentage (%)	**0.5**	**0**

where $1(p_{q \neq Q_b})$ is the indication function and it is equal to zero when $p_{q=Q_b}$ and equal to one, else, $lev_{p,q}(a, b)$ is the distance between the initial a character of p and the initial b characters of q [1].

The LD was calculated by using ten documents with about 750 words from our standard English corpora used as our representative texts. The CDM was then used to generate decoy documents. The percentage of the average LD was computed to be 62.7%. The high value implies that there is a large difference between the original document and the decoy document, which implies the low probability of transforming the decoy document to the original document.

The Levenshtein distance between the number of bit changes in the plaintext data and the decoy message of the CDM is compared with Yoon *et al.* [43], Kim and Yoon [44] and Beaunardeau *et al.* [45] as depicted in Table 4. The following related works were selected because they use similar dataset. Comparing the proposed work with related works using a dissimilar dataset may be inadequate as it reflects comparing oranges to apple which is not plausible

TABLE 3. Sample dataset of the input and output sentences generated during the dtt test.

her teacher is my mentor	she found the tv series box of moon light
find me the rise and fall of the great	incorporate time warp into my schedules
show me the documentary	The teacher ate the cookie
find the schedule for across the line	look for the girl in the mirror
he will visit the new temple today	she travelled with benny
search for the man in my garage	can you find me the book
i will go with everyone	i have played football with john terry
she met him at the coffee shop	i want to read the novel alone this night
where can i buy video game	where can i see the television show falling away from me
where in the neighborhood can i buy oysters	please look up the news press album
i want to listen to the song the loving spirit	find the novel what we did on our holiday
where is road to the stage playing	come to work as soon as possible
our cousin passed his exams	let us attend the ordination ceremony
what animated movies are playing in the area	we travelled for many weeks
what is the movie schedule	what movies are playing at dipson theatres
show me the movie called emerald city	my son passed his exams
i want to know if it is freezing in the attic	i left the bus station
we will be having our lunch at the sea gull today	show me the movie schedule for movies around here
provide me with movie schedules for next month	what time is soul surfer playing
it be hotter in netherlands antilles today	book a seat for my brother
look up the weather at the	i need a table for 8
what is the forecast for saint martin next year in Oregon	make me a reservation in town somewhere nearby for a party of 4
give two stars to current album	what is the 10 pm forecast for Maryland
book a restaurant for 3 on june the fifth	will there be a blizzard in the united states
the atmospheric pressure is normal today	what is the weather like in andorra
show weather forecast for dovre national	book reservations for all of us on jul 6 in hainesville
book verdure serving restaurant in bloom city	i like to organize a party of five today in a cool joint
what is the weather in the city of evangeline	can i have the forecast for 6 am
i would like to take my lunch at home	it be warm in rhode island next week
i want to go to a bistro at a taverna with internet	will it snowstorm in the national wildlife refuge
locate a cafeteria that serves hotdogs	will it get chillier on the 6th of December
can you get the forecast in greensburg	make a booking for a party of 10
it is a stormy day in west virginia	is it going to be warm on november 20

READING

	The highlighted texts are the representative input text.
	The unhighlighted texts are output text (extracts from the decoy document)

TABLE 4. Sample dataset used for the levenshtein distance.

[43]	[44]	[45]	CDM
Plaintext → *hello bob how* *i* *s it going it is* → *simulation of* → *our chatting* → *program alice* → *and bob who* → *shared same* → *secret key can see real plaintext message however malicious user eve will get fake message if eavesdrop their communicatio n message*	Plaintext → *hello bob how* → *is it going it is* → *simulation of* → *our chatting* → *program alice* → *and bob who* → *shared same secret key can see real plaintext message however malicious user eve will get fake message if eavesdrop their communicatio n message*	Plaintext → *hello bob how is it* → *going it is simulation* → *of our chatting* → *program alice and bob* → *who shared same* → *secret key can see real* → *if plaintext message however malicious* *ea user eve will get* *commun fake message* *vesdrop their ication message*	Plaintex t *hello bob how is it* → *going* → *it is* → *simulatio n of our* → *chatting program alice and* → *bob who shared* → *same secret key can see real plaintext message however malicious user eve will get* → *fake message if eavesdrop their communication message*
Decoy Message *table* → *dad bow to is* → *house to be* → *determined is* → *low minister* → *pencils chair* → *but low so* → *stored show builds day try can they examines builder leather delicious book she cant bet wake holiday to destructs share contaminatio n extends*	DecoyMessage *the door* → *opens to the table* → *i don the first the* → *day the been s is* → *a street the* → *continued to s a* → *beat is* → *the phone got a he was the s a second Rachel is toms bedside and summer s in the s not to see you re s i dont was next to starts he could begins to b*	DecoyMessage → *cat is paradise high* → *school football team new* → *jersey big* → *snow isobel spirit* → *the first of two rovers of* → *nas a* → *when her home was hit by a bomb the dog designed the Totzeret HaAretz Tower park in North America when it opened last month*	Deco Message *she comes to school* y *daily where in* → *the world* → *is* → *candice they went* → *down by the riverside send the* → *parcel to mongolia show schedule* → *for harkins theatres* → *where can i watch the game*

as described by [1]. Sample dataset used for the levenshtein distance is shown in Table 4.

D. TEST FOR RANDOMNESS

Aiming to evaluate the proposed system in terms of its relation to cryptography, the proposed CDM was tested using the NIST test suite [47]. Cryptographic applications lay great emphasis on randomness in their constructions. Non-randomness in generated sequences is believed to degrade the security of security systems as it opens an avenue for the success of a cryptanalytic attack. Several suites such as the NIST Statistical.

Test Suite (NIST STS), TestU01, Diehard test and others are battery of statistical test often used for evaluating the security models to detect deviations of the generated sequence from randomness. The NIST STS is an important evaluation suite mostly used for formal approvals or certifications. This study applied the NIST suite to test the randomness of the decoy message generated by the proposed CDM.

TABLE 5. Result for test for randomness.

FILE LENGTH: 1000000			
SIGNIFICANCE LEVEL: 0.05			
Test No.	**Test Name**	**P-Value**	**Result**
1	The Frequency (Monobit) Test	0.534910741	PASS
2	Frequency Test within a Block	0.652031492	PASS
3	The Runs Test	0.439176363	PASS
4	Tests for the Longest-Run-of-Ones in a Block	0.46373944	PASS
5	The Binary Matrix Rank Test	0.517835344	PASS
6	The Discrete Fourier Transform (Spectral) Test	0.725338365	PASS
7	The Non-overlapping Template Matching Test	0.163833939	PASS
8	The Overlapping Template Matching Test	0.266335342	PASS
9	Maurer's "Universal Statistical" Test	0.873538273	PASS
10	The Linear Complexity Test	0.392273636	PASS
11	The Serial Test	0.452922834	PASS
12	The Approximate Entropy Test	0.762242456	PASS
13	The Cumulative Sums (Cusums) Test	0.637383894	PASS

A string of plaintext message was encrypted using the proposed system and repeatedly decrypted using the wrong keys for several times. A random sampling method was used to capture a close to perfect representation of the total population (population, in this case, the decoy message generated). Sampling errors or variations that may result due to sampling is controlled using larger samples (in this case the frequency/number of times of decryption using the wrong keys) which was computed 1,000 times. The NIST STS has about 15 tests for evaluating the randomness of the data sequence. It formulates a hypothesis based on the sequence. In this case, the null hypothesis, Ho assumes that the sequence being tested is random and an alternative hypothesis, Ha, which assumes that the sequence is not random. To draw a conclusion on the test, the resulting p-value of each test is compared with the significance level α.

A p-value lesser/greater than α implies that the hypothesis is rejected/accepted.

The NIST recommends that more test is needed to determine the accuracy and quality of randomness of the bit streams. However, the selection of tests to use depends on the considered data and its application domain. During the set up to run the NIST STS suite, few tests were not considered. For example, the Lempel-Ziv Compression Test considers

TABLE 6. Sample dataset of decoy message generated during the nist test.

he played all the way home	she roamed round the street
my agenda today is to visit the cinema	what movie theatre is playing the wanderer
they cruised in the ship	they saw the movie at megaplex theatres
he meandered round the city	sherry brought the letters
i will go to the photo play tomorrow	the caribbean cinemas plays every weekend
she was surrounded by those men	the photograph displayed so much
please get me the showtimes for films at malco theatres	she looked at him so hard
they watched animated movies in the caribbean cinema	when is the crucible of man playing at the movie house
they go to the island every summer	she was a legend in the world of british theatre
i will be seeing the musketeers tonight	she was precious to us all
he met up with the petite girl at the mall	the lilacs in the spring play daily at the movie house
reserve a seat for us at the white inn	he bought me the picture blink of an eye
he got me office supplies at the stationery	he spent all day learning to play polka
we saw many comedies at the theatres	i will read the novel between the rivers
please look up the novel heroes of annihilated empires	show me the trailer for the knights
they saw the documentary at the century theatres	play me a trailer for the north west passage
he worked at the municipal	the closest movie house is playing animated movies
we will see the tv series octavia	he made the comic strip himself
he ran errands for them yesterday	where can i read the book the omega stone
i want to watch prodigal wife today	show me the movie schedules for movies playing around here today
please get the photograph of my best friend	he took my painting away
we played the collectors video game last night	we want to buy wristwatch at the mall
please provide me with all the movie schedules for next month	we will be seeing the television show titled justice league
she likes to see chick flicks every wednesday night	he bought the lady a diamond necklace
the chateau is built around the courtyard	can i get the showtimes for the closest movie theatre that has the newest films
he cycled in the playground	we found the movie times for fox theatres
the county mall is beside the bridge	we spotted the towering building today
the theatres will show crush and blush tonight	show me the movie schedules for movies playing around here today

TABLE 6. *(Continued.)* **Sample dataset of decoy message generated during the nist test.**

we ate dinner at the fancy cafeteria	we travelled by the bridge of san luis
he memorized the poem for many days	she likes to read the novel good doctor
he bought the picture at warren theatres	i am looking for the birth of a nation

i will buy a neckpiece made of ruby	they like to watch the hola mary footage
we saw the trailer a moment ago	show me the photograph a woman from the street
we want to play the game piety street	i scanned through the novel
she ordered a years subscription for the newsletter	they came when it was dark
they spent their holidays in new york city	what are the movie schedules for movies playing in the neighbourhood
she love reading fiction	can i get the game list of the white singles
we saw the picture bugs bunny	we drove around town
who will tour the city alongside me	where is the beach of lost children playing
find the movie schedules for me	we saw the city of light
find a book called the polish bride	who can find the television series me and my guitar
we bought the atlas at the book store	find the painting searchlights
give me the spirit the earth aflame tv show	i will look up the schedule for the screenplay
find the winter song game	can i see the movie schedules
where can i see animated movies that is nearby	where can i find the creative video game
they looked for the picture of house foundation	can i get the showtimes for feel the passion
when is the trailer for man in blues	what time is class playing at amco entertainment
we saw several movies in paramount theatres	what animated movies are in the neighbourhood
find an interesting movie schedule for them	she followed me to the cinema
show me movie schedules for the movies close by at sunset	what time is the queen of moulin rouge playing
i want to watch strange brother	find the movie schedules for great escape theatres
find my tribute show	can you find me a trailer for phineas redux
what movie schedules are at national amusements	the north dakota tv series will be playing next spring
give me showtimes for films in the neighbourhood	find the schedule for movies at the megaplex theatres
we will eat sushi at the nearest food place	find me the lace and whiskey soundtrack
where do we buy chemicals for the engineers	find films at magic johnson theatres
the 21st century the resurgence is playing now	he played us the album by bridget

TABLE 6. *(Continued.)* **Sample dataset of decoy message generated during the nist test.**

pull up the showings for pop goes the weasel at the theatre	is the couch trip at the nearest cinema
i want to play the game show me the wonder	show me screenplay for tomorrow
we bought so much at the plaza	what movies are playing nearby
it is movie times at national amusements	find the movie schedule for animated movies in the neighbourhood
i want to go to the brown theatre	i want to play the video game the coyote kings of the space
the santikos theatres has several movies lined up for the week	when is the great question playing at the closest movie house
they bought the novel for her	i was fascinated with the view of the citadel
she visits the country's house often	he went surfing for many days

how far the tested sequence can be compressed. Thus, if a sequence can be significantly compressed, then it is judged to be non-random. The proposed CDM does not cater for compression and as such, the Lempel-Ziv test for compression was not considered

Experiments

n - Length of bit string (Total n $>$ 1,000,000 bits)
ε- Sequence of bits as additional input supplied by testing code

M- Length of each block

Predefined Significance Level $\alpha = 0.05$

Predefined Block frequency M and m, 128 and 9

String of Plaintext:

$\rightarrow$ *we watched the movie emerald city*
$\rightarrow$ *i want to read the black book*
$\rightarrow$ *they travelled round the world*

Table 5 depicts the result from the test. More details can be found in Appendix B.

For each of the 13 tests carried out, the randomness test is considered to have passed as the p-value is greater than the significance level (p-value $> \alpha$). In this case, we accept the null hypothesis H_o to mean our generated bit stream is random. Passing all the battery of test indicates that the generated sequence is random with no obvious statistical defects. Sample set from the decoy message collected is given in Table 6.

Additionally, a significance test was carried out to evaluate the proposed CDM in terms of its entropic property. The test was carried out to check the degree of randomness when a correct key and when several incorrect keys are used to decrypt the plaintext. The test was carried out using MATLAB 2013(a). A plaintext was encoded using the proposed model. In the first instance, decryption was done using the correct key, yielding the Plaintext message.

TABLE 7. Functional comparison of related work with the proposed cdm.

Features	Karuna et al., [25]	Whitham [23]	Karuna et al., [13]	Proposed CDM
Adaptability to Various Domains	X	X	X	√
Quality of Knowledge (Completeness and correctness)	X	X	X	√
Syntactic Cohesion	√	√	√	√
Semantic Coherence	X	X	√	√
Non-detectability of Partial Information in the Source Document	X	X	X	√

In the other part, decryption was done repeatedly using random keys for 1,000 times.

P represents the Plaintext, P* represents the generated decoy messages.

String of plaintext

→ *they travelled round the world*
→ *I want to read the black book*
→ *we watched the movie emerald city*

The first and second hypotheses are as follows,

H_o = There is no difference in the entropy between P and P*

H_a = There is a difference in the entropy between P and P*

To put it in a simpler context, Ho implies that the plaintext P and the decrypted decoy message P* cannot be distinguished while Ha implies that they can be distinguished. The entropic distribution of the wrong plaintext P* represents the test statistics while the entropy of the plaintext P represents the observed value. The significance level used was 0.05. Figure 4 depicts the result for the test of the entropic property of the proposed encoder.

A small P-value compared to the 0.05 significance level indicates that the observed data P is not included in the scope of P*. Thus, H_a is rejected. This implies that the adversary cannot predict or acquire the plaintext from the distribution of the ciphertext during the attack as there is a difference

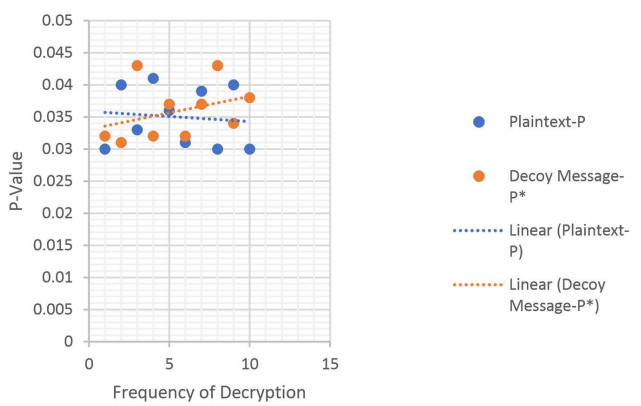

FIGURE 4. Experiment for the entropic property of the plaintext, P and decoy message, P*.

between the entropy of H_o and H_a. The statistical test considered word level security, sentence level security where decoys form meaningful messages which are semantically and syntactically coherent was evaluated using the DTT Test described in Section V.

E. FUNCTIONAL COMPARISON OF THE PROPOSED CDM WITH RELATED WORKS

The comparison of different models with each other is often challenging due to how a particular model process and train data may be different from the other model being compared to it. Additionally, there is no numerical metric of a benchmark to compare models with [41]. Thus, a function-based approach of comparison was adopted for this work. Table 7 shows a functional comparison of the work by [25] which is related to this study. Functional comparison of related work with te proposed CDM is presented in Table 7.

F. LIMITATIONS AND PROPOSITION FOR FUTURE WORKS

Another Evaluating language models is widely considered to be a challenge in its own right. This difficulty is based on the premise that using automatic metrics may fail to correspond with human perceptions of language, thus enforcing the evaluation to be done manually using human subjects which may be subjective. A group of human judges may perceive language in another way from another group.

Evaluating the domain-specific model is restricted to the domain explored during the implementation of the proposed model. This is on the grounds that it is challenging to investigate the entire elements in every field/domain which may be humongous. Additionally, evaluating some specific domain-based texts requires an expertise human-level of comprehension and foundational knowledge which our tentacles of research constrains us from reaching.

Given that our proposed CDM model has been designed to cater for all domains, future works may consider exploring text documents in all domains and feeding the keywords into our model. This should be followed by getting experts in all

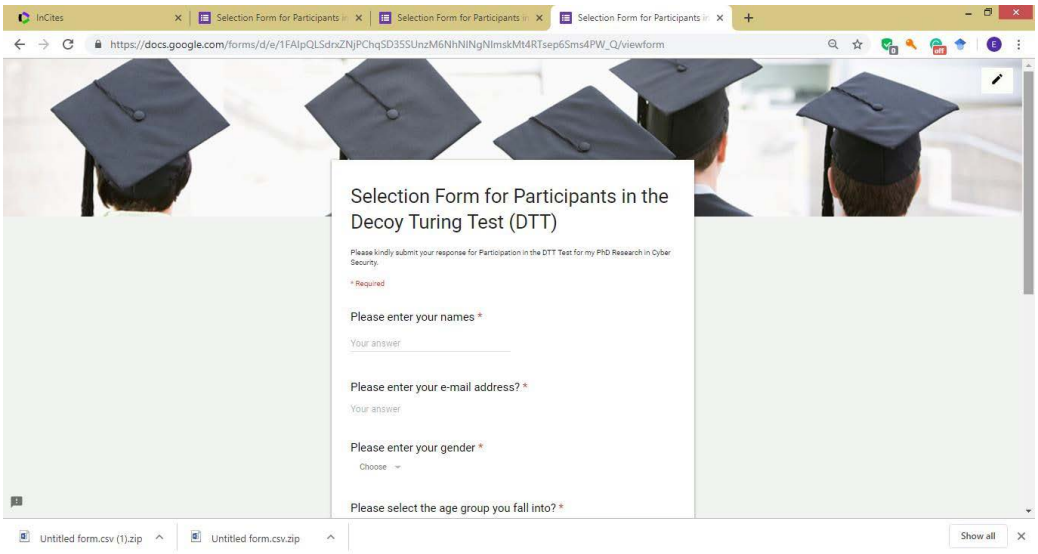

FIGURE 5. Result of selection.

such domains to conduct a real-world experiment. While this may require much effort, it is not impossible and it will be a significant contribution to the deception system from the global domain perspective.

In conclusion, as noted by [48], research is required in the area of defining a clear methodology for testing, evaluating and subsequently, benchmarking deception systems which is a big shortcoming and is lacking in the field of deception systems given the potentiality it can provide in this surging period of cyber attacks.

VI. CONCLUSION

The staggering statistics of the frequency of data theft and compromise have put the global world under unrest. Several data breaches have been reported where advanced attackers bypass the radar of traditional control measures of enterprise networks to exfiltrate and steal confidential documents. Additionally, cybercriminals can carry out business email compromise after they have penetrated the network. The use of decoys and deception is one of the numerous solutions used to protect confidential documents after a cybercriminal infiltrates the network. Deception measures involve seeding a network system with information that appears legitimate but is in fact fake and misleading. It boosts the security of network systems and components by leveraging strategies of deceit, denial, camouflage, misinformation and obfuscation.

This paper presents a cognitive deception model (CDM) which concentrates on cognitively misleading and burdening the cyber attackers by wasting their time, effort and resources. It takes the messages in the original document as input and generates syntactically cohesive and semantically coherent independent looking but plausible and convincing decoy messages to burden and deceive the adversaries cognitively. The CDM produces decoy documents that does not only focus on the textual characteristics but also on the completeness and correctness of the messages. It embodies the critical criteria of deception systems as it repackages, masks and mimics the real document.

We also made a major contribution to the literature on the decoy and deception system by introducing domain-specific decoys. Domain-specific decoys can be applied to any domain as it helps to restrict decoys to a specific domain to further improve the capacity of deception in the system. For instance, a cybercriminal exfiltrating data on the weather forecast domain will fail to be convinced/misled when the document he retrieves is based on a restaurant's domain.

Additionally, the proposed CDM completely changes the entire document which handles the flaws of some of the current approaches where the attackers may learn partial information from the original message and may then be able to reconstruct the message.

Finally, the proposed CDM contributes to the literature in curtailing business email compromise, as it can be implemented as a plugin in email systems to mislead attackers during the process of trying to read legit emails of employees of an organization.

Future works may concentrate on improving the model to incorporate decoy images and other artefacts that are usually included in documents. Additionally, newly introduced transformer models such as BERT, GPT [49], [50] amongst others may be considered as a base to build or enhance the deception model in future works.

COMPLIANCE WITH ETHICAL STANDARDS

CONFLICTS OF INTEREST

All authors declare that there is no conflict of interest.

HUMAN AND ANIMAL'S RIGHTS

No studies with human participants or animals performed by any of the authors.

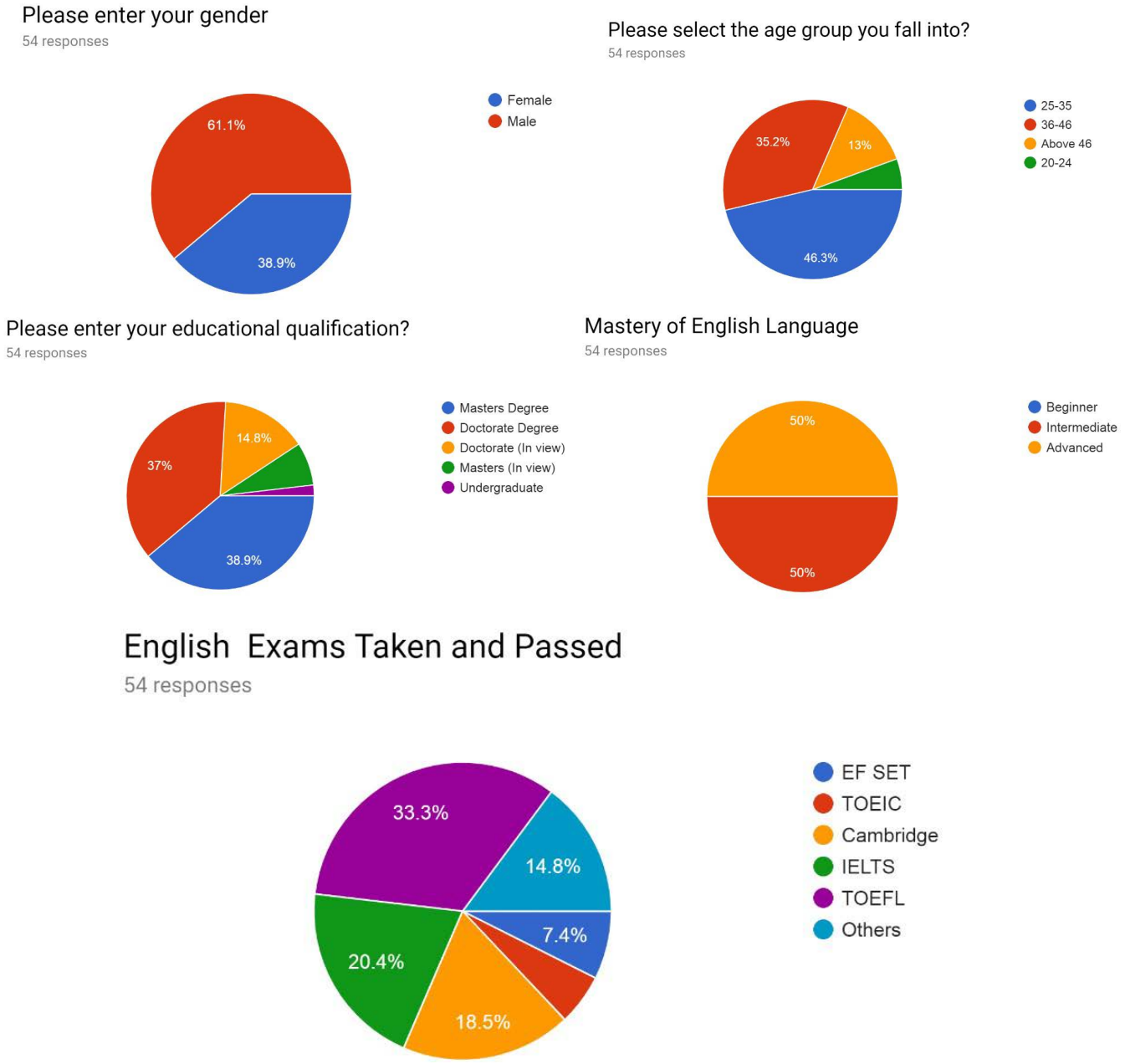

FIGURE 6. Result of the selection of human judges.

INFORMED CONSENT

In this article, informed consent was not required as no human or animals were involved.

APPENDIX A: DETAILS ON THE SELECTION OF HUMAN JUDGES

See Figures 5 and 6.

APPENDIX B: IMPLEMENTATION OF THE NIST STATISTICAL TEST SUITE

Data input was supplied as a stand-alone program using files of arbitrary length. Files contained binary data where each byte contains eight bits worth of 0's and 1's where each word is the standard 2 bytes or 16 bits. A python implementation of the suite using a GNU public license software sp800_22_tests was used for executing the program. The implementation provides a separate python file to read the file containing the binary data file for each test with a summarized result at the end. Implementation software can be found at https://github.com/dj-ongithub/sp800_22_tests.

The following symbols are the inputs to the suite as applied to the data file.

- n - Length of bit string (Total n > 1,000,000 bits)
- ε- Sequence of bits as additional input supplied by testing code
- M- Length of each block

- Predefined Significance Level $\alpha = 0.05$
- Predefined Block frequency M and m, 128 and 9

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license ()" for more information.
>>>
RESTART: C:\ Users\Esther\Python\sp800_22 _tests-master\sp800_22_testsmaster\sp800_22 _serial_test.py
>>> SUMMARY
------
monobit_test 0.534910741182 PASS frequency_ within_block_test 0.652031492431 PASS runs_test 0.439176363136 PASS longest_run_ones_in_a _block_test
0.463739440251 PASS binary_matrix_rank_test 0.517835343612 PASS dft_test 0.725338365124 PASS non_overlapping_template _matching_test 0.163833939303PASS overlapping _template_matching_test 0.266335341899 PASS
maurers_universal_test 0.873538272657 PASS
linear_complexity_test 0.392273635589 PASS
serial_test 0.452922833636 PASS
approximate_entropy_test 0.762242456391 PASS
cumulative_sums_test 0.637383893939 PASS
```

REFERENCES

[1] A. E. Omolara, A. Jantan, O. L. Abiodun, K. V. Dada, H. Arshad, and E. Emmanuel, "A deception model robust to eavesdropping over communication for social network systems," *IEEE Access*, vol. 7, pp. 100881–100898, 2019.

[2] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," in *Cyber Deception*. Cham, Switzerland: Springer, 2016, pp. 23–50.

[3] N. H. Chowdhury, M. T. P. Adam, and T. Teubner, "Time pressure in human cybersecurity behavior: Theoretical framework and countermeasures," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101931.

[4] (2021). [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/2021/year-in-review-2021/

[5] (Dec. 19, 2021). *2021 Data Breach Investigation Reports*. Data Breach Statistics by Industry. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/2021/data-breach-statistics-by-industry/

[6] A. Cidon, L. Gavish, I. Bleier, N. Korshun, M. Schweighauser, and A. Tsitkin, "High precision detection of business email compromise," in *Proc. 28th USENIX Secur. Symp. (USENIX Security)*, 2019, pp. 1291–1307.

[7] S. Mansfield-Devine, "The imitation game: How business email compromise scams are robbing organisations," *Comput. Fraud Secur.*, vol. 2016, no. 11, pp. 5–10, Nov. 2016.

[8] C. Cross and R. Gillett, "Exploiting trust for financial gain: An overview of business email compromise (BEC) fraud," *J. Financial Crime*, vol. 27, no. 3, pp. 871–884, Apr. 2020.

[9] Y. Mao, Y. Zhang, M.-R. Chen, Y. Li, and Y. Zhan, "Efficient attribute-based encryption schemes for secure communications in cyber defense," *Intell. Autom. Soft Comput.*, vol. 22, no. 3, pp. 397–403, Jul. 2016.

[10] E. O. Abiodun, A. Jantan, O. I. Abiodun, and H. Arshad, "Reinforcing the security of instant messaging systems using an enhanced honey encryption scheme: The case of WhatsApp," *Wireless Personal Communications*, vol. 112, no. 4, pp. 1–24, 2020.

[11] J. Szefer, P. Jamkhedkar, D. Perez-Botero, and R. B. Lee, "Cyber defenses for physical attacks and insider threats in cloud computing," in *Proc. 9th ACM Symp. Inf., Comput. Commun. Secur.*, Jun. 2014, pp. 519–524.

[12] H. Tang, Q. T. Sun, X. Yang, and K. Long, "A network coding and DES based dynamic encryption scheme for moving target defense," *IEEE Access*, vol. 6, pp. 26059–26068, 2018.

[13] P. Karuna, H. Purohit, S. Jajodia, R. Ganesan, and O. Uzuner, "Fake document generation for cyber deception by manipulating text comprehensibility," *IEEE Syst. J.*, vol. 15, no. 1, pp. 835–845, Mar. 2021.

[14] P. Karuna, "Manipulating comprehensibility of text: An automated approach to generate deceptive documents for cyber defence," Ph.D. dissertation, Volgenau School Eng., Fairfax County, VA, USA, George Mason Univ., 2019.

[15] K. E. Heckman, F. J. Stech, B. S. Schmoker, and R. K. Thomas, "Denial and deception in cyber defence," *Computer*, vol. 48, no. 4, pp. 36–44, 2015.

[16] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 145–160.

[17] A. E. Omolara, A. Jantan, O. I. Abiodun, H. Arshad, K. V. Dada, and E. Emmanuel, "HoneyDetails: A prototype for ensuring patient's information privacy and thwarting electronic health record threats based on decoys," *Health Informat. J.*, vol. 26, no. 3, pp. 2083–2104, Sep. 2020.

[18] M. H. Almeshekah, "Using deception to enhance security: A taxonomy, model, and novel uses," Purdue Univ., West Lafayette, IN, USA, Tech. Rep., 2015. [Online]. Available: https://docs.lib.purdue.edu/open_access_dissertations/1334

[19] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *Proc. Int. Conf. Secur. Privacy Commun. Syst.* Berlin, Germany: Springer, 2009, pp. 51–70.

[20] A. Juels, "A bodyguard of lies: The use of honey objects in information security," in *Proc. 19th ACM Symp. Access control models Technol. (SACMAT)*, 2014, pp. 1–4.

[21] M. B. Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2011, pp. 181–200.

[22] S. Rauti and V. Leppanen, "A survey on fake entities as a method to detect and monitor malicious activity," in *Proc. 25th Euromicro Int. Conf. Parallel, Distrib. Netw.-based Process. (PDP)*, 2017, pp. 386–390.

[23] B. Whitham, "Automating the generation of enticing text content for high-interaction honeyfiles," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 1–10.

[24] P. Karuna, H. Purohit, O. Uzuner, S. Jajodia, and R. Ganesan, "Enhancing cohesion and coherence of fake text to improve believability for deceiving cyber attackers," in *Proc. 1st Int. Workshop Lang. Cognition Comput. Models*, 2018, pp. 31–40.

[25] P. Karuna, H. Purohit, R. Ganesan, and S. Jajodia, "Generating hard to comprehend fake documents for defensive cyber deception," *IEEE Intell. Syst.*, vol. 33, no. 5, pp. 16–25, Oct. 2018.

[26] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1378–1387.

[27] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 160–167.

[28] L. Deng and Y. Liu, *Deep Learning in Natural Language Processing*. Singapora: Springer, 2018.

[29] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.

[30] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1045–1048.

[31] G. Fan, X. Diao, H. Yu, K. Yang, and L. Chen, "Software defect prediction via attention-based recurrent neural network," *Sci. Program.*, vol. 2019, pp. 1–14, Apr. 2019.

[32] A. Demi. (2020). *How BERT and GPT Models Change the Game for NLP*. (Feb. 3, 2022). [Online]. Available: https://www.ibm.com/blogs/watson/2020/12/how-bert-and-gpt-models-change-the-game-for-nlp/

[33] P. Stuart. (2019). *Trends in Natural Language Processing*. (Feb. 3, 2022). [Online]. Available: https://www.elderresearch.com/blog/trends-in-natural-language-processing/

[34] G. Shradda. (2021). *Why Transformers are Increasingly Becoming as Important as RNN and CNN?*. (Feb. 3, 2022). [Online]. Available: https://analyticsindiamag.com/why-transformers-are-increasingly-becoming-as-important-as-rnn-and-cnn/

[35] S. Narang, H. W. Chung, Y. Tay, W. Fedus, T. Fevry, M. Matena, K. Malkan, N. Fiedel, N. Shazeer, Z. Lan, Y. Zhou, W. Li, N. Ding, J. Marcus, A. Roberts, and C. Raffel, "Do transformer modifications transfer across implementations and applications?" 2021, *arXiv:2102.11972*.
[36] S. Chua and N. Kulathuramaiyer, "Semantic feature selection using WordNet," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Sep. 2004, pp. 166–172.
[37] K. Zhang, J. Sun, and B. Wang, "A WordNet-based approach to feature selection in text categorization," in *Proc. Int. Conf. Intell. Inf. Process.* Boston, MA, USA: Springer, 2004, pp. 475–484.
[38] G. A. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.
[39] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 5–13.
[40] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.
[41] O. Ludwig, X. Liu, P. Kordjamshidi, and M.-F. Moens, "Deep embedding for spatial role labeling," 2016, *arXiv:1603.08474*.
[42] I. B. A. Turing, "Computing machinery and intelligence-AM Turing," *Mind*, vol. 59, no. 236, p. 433, 1950.
[43] J. W. Yoon, H. Kim, H.-J. Jo, H. Lee, and K. Lee, "Visual honey encryption: Application to steganography," in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2015, pp. 65–74.
[44] J.-I. Kim and J. W. Yoon, "Honey chatting: A novel instant messaging system robust to eavesdropping over communication," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2184–2188.
[45] M. Beunardeau, H. Ferradi, R. Géraud, and D. Naccache, "Honey encryption for language," in *Proc. Int. Conf. Cryptol. Malaysia*. Cham, Switzerland: Springer, 2016, pp. 127–144.
[46] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages," in *Proc. LREC*, vol. 29, 2012, pp. 31–43.
[47] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, and A. Heckert, "NIST special publication 800–22: A statistical test suite for random number generator for cryptographic applications," National Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2001.
[48] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2019.
[49] S. Yu, J. Su, and D. Luo, "Improving BERT-based text classification with auxiliary sentence and domain knowledge," *IEEE Access*, vol. 7, pp. 176600–176612, 2019.
[50] M. V. Koroteev, "BERT: A review of applications in natural language processing and understanding," 2021, *arXiv:2103.11943*.

OLAYIWOLA TOKUNBO TAOFEEK received the Ph.D. degree from the School of Computer Science, Universiti Teknologi MARA (UiTM), Malaysia. His research interests include computer science, cybersecurity management, information technology, project management, and data mining.

MOATSUM ALAWIDA received the B.Sc. degree from Mutah University, Jordan, in 2005, the M.Sc. degree in information systems from The University of Jordan, in 2010, and the Ph.D. degree from the School of Computer Sciences, Universiti Sains Malaysia. His research interests include chaotic systems, chaos-based applications, multimedia security, cryptography, and national security.

ABDULATIF ALABDULATIF (Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from RMIT University, Australia. He is currently an Assistant Professor with the College of Computer, Qassim University, Saudi Arabia. His research interests include applied cryptography, cloud computing, and data mining.

ABIODUN ESTHER OMOLARA received the bachelor's, master's, and Ph.D. degrees in computer science from the School of Computer Sciences, Universiti Sains Malaysia. She is currently with the Department of Computer Science, University of Abuja, Nigeria. Her research interests include computer vision, cryptography, information and communication security, artificial intelligence, natural language processing, network and communication protocol, cybersecurity, digital forensics, and the IoT security. Others are application development, database, web design, and web application.

OLUDARE ISAAC ABIODUN received the Bachelor of Technology degree in computer science and mathematics from the Federal University of Technology Minna, Nigeria, the Master of Technology degree in information and communication technology from the Federal University of Technology Owerri, Nigeria, the Ph.D. degree in nuclear and radiation physics from the Nigerian Defence Academy, Kaduna, in 2012, and the Ph.D. degree in computer science from the Universiti Sains Malaysia, in 2020.

He received funding support for his research from the Center for Cyber Safety and Education, United States Internal Revenue segregated fund of $(ISC)^2$, Inc., through the $(ISC)^2$ graduate cybersecurity scholarship award. His research interests include cybersecurity, digital forensics, terrorism, national security, computer vision, cryptography, information and communication security, artificial intelligence, optimization, nuclear security, network and communication protocol, and the IoT security.