

On the Problem of Features Variability in Sequence Learning Problems

Mohammed Onimisi Yahaya

Abstract—Sequential learning problems such as speech, cursive handwriting, time series forecasting and protein sequence prediction. Both Speech and cursive handwriting recognition are challenging problems to Pattern recognition systems, in particular speech signal. Some peculiar characteristics of these types of problems are that, the signal or pattern evolves with time, modeling a long time dependencies in this pattern is a major challenge. Hidden Markov models (HMM) have been applied for these types of problems. Due to some obvious shortcomings of HMM, neural networks were also explored and applied as well as their hybrids. The problem of feature variability in sequence learning is still a challenging problem. In this paper, we analyzed the problem, present some methods in feature variance suppression in character recognition, and review some research efforts in modification of neural networks and applications. We proposed a structure for a state-based neural network.

Index Terms—Sequence learning, feature variability, neural network.

I. INTRODUCTION

Sequential learning problems are type of problems in which the data or signal of interest are sequential in structure [1]. Examples of such problems are control of attention such as human pattern recognitions, sequence recognition problems such as speech, cursive handwriting, time series forecasting and protein sequence prediction. Others are motor control such as memory for partially observable words. In fact, virtually every real world task is sequential problem. Both Speech and cursive handwriting recognition are challenging problems to Pattern recognition systems, in particular speech signal. One peculiar characteristic of these types of problems are that, the signal or pattern evolves with time, modeling a long time dependencies in this pattern is a major challenge. Concerted effort by early researchers to represent time explicitly in the pattern did not yield an interesting result, For example, symbolic approaches such as grammar learning is not useful when real valued noisy features such as speech signals are involved.

In this paper, we will concentrate on the analysis and review of challenges, research effort, tools that have been applied particularly to two of the above mentioned problems-speech and handwriting recognitions, of particular interest is the problem of recognizing variable length feature of the same speech signal. Hidden Markov Models (HMM)

have been applied in providing solutions to these classes of problems. For examples see [2], [3]. In HMM, the patterns are presented as set of frames of sliding windows. It was originally applied for Speech Recognition, it success led to other application in character and cursive hand writing recognition, for examples see [4]. Due to some obvious shortcomings of HMM such as the discrete form of HMM cannot store real values, the requirement of large data set for training and lack of good algorithms to learn or training some of it topologies. Researchers turned attention to neural networks, which have excellent discriminative capability and approximation of complex functions, but fails in time variance, long term dependency modeling, though there are some few applications of static neural networks in speech recognition, see [5] for recent application. Hybrids systems of HMM and ANN are explored to combine the capabilities of both models in Speech Recognition. See [6] for a comprehensive survey of on HMM/ANN hybrids. Till date speech recognition is still a difficult problem, especially feature variance.

The remaining part of the paper is organized as follows: In Section II, we present the description of the problem of feature variance in pattern recognition system. Section III presents the theory and literature survey on the proposed framework of a state based neural networks. In Section IV, we reviewed and presented some extension to neural network. Our proposed model of neural network is described in Section V and Section VI is discussions and conclusions

II. ON THE PROBLEM OF FEATURES VARIANCE IN SEQUENCE LEARNING PROBLEMS

In this section, we give a vivid description of the problems of feature variability in speech and text recognition systems. Pattern recognition systems is an aspect of machine learning that have been applied in real life problems such as Speech Recognition, Character/Text Recognition, Bioinformatics, Biometrics etc. Most of these aforementioned domains have their peculiar characteristics, but fundamental to all are appropriate features identification, extraction and representation. Representation and scaling of features is one the greatest challenges in sequence learning problems.

The importance of good Features extraction and representation, as well as good classifiers is critical to the performance and accuracy of Pattern Recognition system. One of the major challenges faced by researchers in this discipline is the different notion of features to different problems that has evade the quest by practitioners and researchers in pattern classification to design a generic model for most of the problems. Feature invariability to scaling,

Manuscript received September 14, 2014; revised January 9, 2015.

The author is with College of Computer Science and Engineering, Affiliated Colleges at Hafr-Al-Batin, King Fahd University of Petroleum and Minerals, Hafr Al-Batin 31991, Saudi Arabia (e-mail: mdonimisi@kfupm.edu.sa).

rotation, translation and occlusion [7], are critical requirement to the success of pattern Recognition systems. Let illustrate these problems of Feature variance with examples. Though, feature variance problems are common to various applications of Pattern recognition. In this paper, we only considered the problem in Speech and text based recognition problems.

Example 1. Variance in Speech Features

Consider the pronunciation of the same English word ‘taken’ in which the elongation is at different syllable such as “taaaken”, taken. They represent same word but they are pronounced differently by different speakers or same speakers at different conditions. This variation may occur as a result of wide range of factors such as pitch, speed of speaker, duration. Generally speech variations occur due to factors such as speaker physiology, rate of speech, children speech, noise, and foreign or regional accent [8]. How can we model this slight difference in our classifier models?

Example 2. Variance in image Features. Consider the English letter “A” of different sizes (see Fig. 1 (a), (b), (c)).

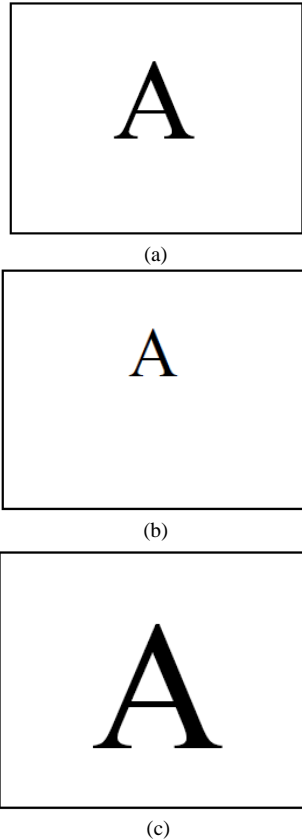


Fig. 1. Variable image sizes of alphabet (A).

A. Techniques of Feature Invariance Suppression in Text/ Character Recognition

We have shown with examples in Section II, Features variance problems in Text/ Character Recognition. There are several approaches employed to suppress this variance in features to improve performance and classification accuracy of the system. In Arabic character Recognition system for instance, see [9], preprocessing of input feature is mainly used to solve the problem of variability in feature. The features are extracted from images, preprocessing techniques such as normalization, thinning, edge detection, baseline detection thinning [10], are used to suppress variance in

features. Segmentation of images especially in cursive text as in Arabic texts, are applied, this method is time consuming and error prone [10]. Segmentation-free methods used HMM based classifier to solve the problem of feature variance such as stroke length elongation. The ability of HMM to model feature vectors as a function of independent variable [3], which emulate the use of sliding windows in speech recognition made several researchers to applied it in text Recognition, where feature variance exist. The performance of this system attested to the capability of HMM in solving the problem of feature variance. Structural features such as vertical projection, contour, horizontal projection, blobs, strokes, primary segment, secondary segments are used to circumvent the problem of feature variance [11]. Yann, *et al.* [12] addressed the problem of invariance in images using convolutional neural networks, where the actual pixel of the images is used as the input to the network. The variance in size of the images is implicitly suppressed by the network.

III. THEORY AND LITERATURE SURVEY

Definition 1: Finite State Automata: A model of simple computing device that can act as language acceptor, It has a finite set of state, hence, it is also referred to as finite automaton or Finite state machine[13]. Finite State machines are categorized into Mealy and Moore Machines. Mealy machines are finite-state machines that act as transducers or translators, taking a string from an input alphabet and producing a string of equal length on an output alphabet [14]. The only difference from Mealy is that Moore is a state emission machine. Which means that symbols are output after the transition to a new state is completed, and the output symbol depends only on the state just reached, that is, $\gamma: Q \rightarrow \Gamma$.

For any given Mealy machine, an equivalent Moore machines can be constructed

Formally, a Mealy machine is a six-tuple

$$M = (Q, \Sigma, \Gamma, \delta, \gamma, q_1)$$

where

- $Q = \{q_1, q_2, \dots, q_{|Q|}\}$ is a finite set of *states*;
- $\Sigma = \{\delta_1, \delta_2, \dots, \delta_{|\Sigma|}\}$ is a finite *input alphabet*;
- $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_{|\Gamma|}\}$ is a finite *output alphabet*;
- $\delta: Q \times \Sigma \rightarrow Q$ is the *next-state function*, such that a machine in state q_j
- after reading symbol δ_k , moves to state $\delta(q_j, \delta_k) \in Q$.
- $\gamma: Q \times \Sigma \rightarrow \Gamma$ is the *output function*, such that a machine in state q_j after reading δ_k symbol writes symbol $\gamma(q_j, \delta_k) \in \Gamma$; and $q_1 \in Q$ is the *initial state* in which the machine is found before the first symbol of a string is processed.

Definition 2: A *deterministic finite automaton* (DFA) [14] is a special type of Moore Machine. It is a quintuple $(Q, \Sigma, \delta, s, F)$, where

- 1) Q is a finite set called the states,
- 2) Σ is a finite set called the alphabet,
- 3) $\delta: Q \times \Sigma \rightarrow Q$ is the transition function,

- 4) $s \in Q$ is the start state, and
- 5) $F \in Q$ is the set of accept states (also called final states).

The Finite State Machine can be viewed in abstract form as a feedback system view that present the functionality of a machine into two functions: the next-state or state-transition function F , and the output function G (see Fig. 2).

- F : States x Symbols \rightarrow States state-transition function
- G : States x Symbols \rightarrow Symbols output function

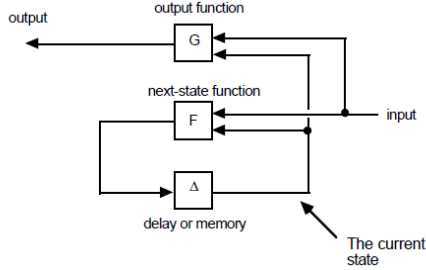


Fig. 2. Feedback view of a finite state machines[15].

Definition 3: Alphabet, string, and Language [13]

Alphabet: a finite set of symbols, denoted by Σ for example, if $\Sigma = \{0, 1\}$ This has 2 symbols, for binary strings. $\Sigma = \{a, b, c, \dots, z\}$, this has twenty six (26) symbols, for English and $\Sigma = \emptyset$, could be empty; has zero symbols. A **string** on Σ : is a finite sequence of *zero or more* symbols from Σ . Example: If $\Sigma = \{a, b, c\}$, then *abbbc* is a string on Σ , and *aa* is another string. If $\Sigma = \{0, 1\}$, then 001 and all binary codes are strings on Σ . A **language**, L , is a set of strings on an alphabet Σ . Thus, $L \in \Sigma^*$. Example: $L_1 = \{a, aaa, abba\}$ is a language defined over $\Sigma = \{a, b\}$. The Kleene closure operator $\{*\}$ [16], is an infinite set of all the strings constructed over set of strings in a regular language. **Definition 2: A formal grammar** is a quadruple (V_N, V_T, P, S) , [16] where, V_N is a set of non-empty non terminals, V_T is a set of non-empty terminals, S where is the start symbol, and P is a nonempty set of production rule. Repeated application of the rewriting rules generates a specific and terminal symbols, respectively, and is a set of generates a specific that belong to the language. Chomsky and Schützenberger [17] categorized formal languages into four types: Type 0: regular, Type 1-context-free, Type 2-context-sensitive, and Type 3-phrase structure, each category is a superset included in its predecessor (e.g., the set of regular languages is strictly included in the set of context-free languages). The classes are defined by the type of production rules allowed in the grammar. Regular languages, generated by regular grammars, represent the smallest class of formal languages in the hierarchy.

IV. HIDDEN MARKOV MODELS (HMM)

Markov Models can be viewed as SFSA with the following characteristics: a probabilistic transition function and a deterministic emission function depending on the current state. It has a memory less property, in that, the probability of observing the automaton in the state q at time t depends only on the state of the model at time $t - 1$. Hidden Markov Model is an extension of Markov Chain and a variant of finite state machine, in which the system modeled, is assumed to be

stochastic with hidden states. Transitions between states are not deterministic as in FSA, hence HMM can be viewed as a Stochastic Finite State Machine in which the transitions depend on Markov process [18]. Hidden Markov Model (HMM) is one of the most successful and popular model for temporal classification. It has been applied to solve sequential problems such as speech, handwriting, Gesture recognition, part of speech tagging and Bioinformatics. Markov Model rich theoretical frameworks provide the background training algorithms for this restricted form, HMM. HMM differs from Markov model into different features[19], (1) It is not possible to observe the state of the model, i.e. q_t is usually not given, (2) the emission function is probabilistic. For excellent tutorials on HMM, see [20].

V. NEURAL NETWORKS

Artificial Neural Network (ANN) is a parallel processor that attempt to mimic the processing paradigm of human brain. The basic processing element is a neuron, (See Fig. 3). ANN is capable of complex function approximation, it learn the relationship between a given input and output without prior knowledge of the problem. It excellent discriminating and learning potential made it one of the most successful computational intelligence tool that has application over a wide range of discipline, including pattern recognition and signal processing. The black box nature of ANN made it fails in some applications. For example, ANN fails as a general framework for speech recognition, due to its inability to manage varying time models [21]. These shortcomings of ANN led to various modification and research on the other forms of ANN to solve the problem of temporal pattern Recognition. For a detailed description of ANN, see [22].

A. Model of a Neuron

The neuron is the basic unit of processing in ANN, an analogy of the synapse in biological neuron. It takes in a set of inputs $\{x_1, x_2, \dots, x_d\}$ and carry out mathematical processing to produce the output. Each neuron is connected with a weighted link that signifies the strength of connection. The output is the weighted sum of the input features acted upon by the activation function. Usually, a node has a set of inputs but one output.

$$f = \sum_{i=1}^d w_i x_i + b$$

B. Neural Networks Topologies

The architecture of ANN indicates the way of ordering and organization of the connections between the input layer, hidden layer and output layer. Researchers have arranged the connections of ANN to solve their respective problems. The most interesting thing is that the choice of topology is problem dependent, though there are some topologies that are widely applied. These are Feed forward, Recurrent and time delay neural networks [23]. This is also refers to as tap delay line (TDL), the input pattern are controlled with a delay in time. In [12], the author described TDNN as a fixed sized convolutional network that shares weight along a single temporal dimension. This type of NN are valuables for class

of problems in which the temporal information as well as the relationship between the input and output sequences are required. It has been applied in various applications, amongst the lots are signature verification [24], character recognition [25], and spoken word recognition [23].

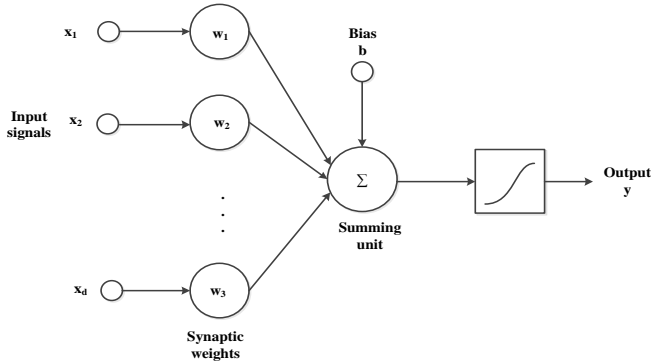


Fig. 3. Model of a neuron.

C. Neural Network and Some Extensions

A recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle [26]. This allows the network an internal state or memory which enables it to exhibit dynamic temporal behavior. RNN has all the terms same as feed forward neural networks but with additional term that indicate the state of the system i.e feedback to the system. The network present behaviour is dependent on it past behaviour. RNN is a linear and nonlinear system; it can be both implemented in hardware and simulated as software. Two of its distinguishing properties are that of time and signal representation. It can be discrete and continuous time system as well as real valued (analog) and quantized (digital) signal. It can be applied as both associative memory tool and sequence mapping tool. In this paper, we will only concentrate on RNN for the sequence mapping and some of its extension. For a guide and comprehensive survey on RNN, see [27]. In sequence mapping using ANN, different input pattern at each time step. The internal memory process the arbitrary sequence of inputs, the entire sequence generated is considered the result of the computation. Sequence mapping RNN have been applied in the following applications, time series prediction, sequence classification such as speech [28], feature extractions and modeling FSM [29].

There are some extensions to static neural network and Recurrent Neural network that we will present briefly. In [30], the author presented Echo State Network, a variant of RNN with sparsely connected hidden layers. The weights in these connections of the hidden neurons are fixed and randomly assigned; this enables the network to learn the weight of the neuron and produce temporal pattern. Elman network [31], a three layer network is used with additional copy context layer to the input layer. There are connections from this copy layer to the input layer. Jordan Network is a variant of Elman with the context layer fed from the output layer instead of the hidden layer. The context layer node has a unique recurrent connection to itself [32]. Hochreiter and Schmidhuber [33] presented an architecture called Long Short Term Memory. This performed better than traditional RNN as the network can learn from experience given appropriate input weight matrix to classify and predict time series in a long time

variance sequence. It has outperformed HMM and RNN as a sequence learning method in some applications, such as in unsegmented cursive hand writing [34] and speech applications [35]. Its architecture is made up of RNN + LSTM blocks which augment the network by remembering arbitrary value in a long period of time. The LSTM block is made up of gates that determine the significance of input to remember, when to remember, when to forget, how long and when to produce output.

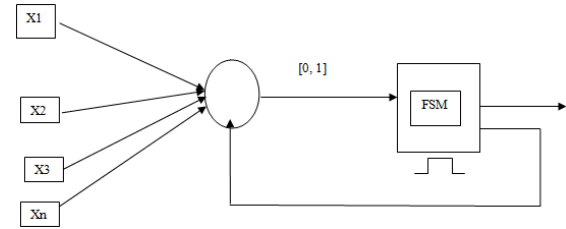


Fig. 4. Model of a state-based neuron.

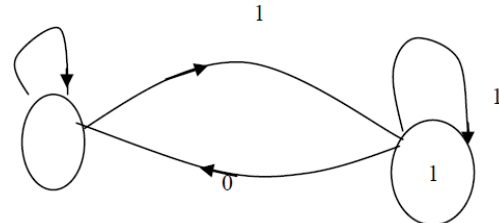


Fig. 5. The transition diagram of the FSM.

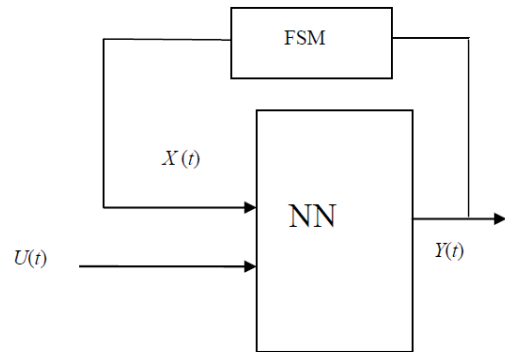


Fig. 6. Block Structure of the state-based neural networks.

VI. A PROPOSED FRAMEWORK FOR STATE-BASED NEURAL NETWORK

A static neural network uses current information at each time to make classification, while RNN learns the dynamic changes. The states from architectural point of view can be model into forms; State-space model and input-output model. Consider a system input vector at time t as $U(t)$, The output vector is $y(t)$, and State of the system is $x(t)$. The general model will be given as

$$X(t+1) = F(x(t), U(t))$$

$$Y(t) = G(x(t), u(t))$$

The state of a system is the past of a system that is relevant to the future behavior [36]. Any discrete time systems can be modeled with the above equation. The proposed state-based

neuron is as shown in Fig. 4, the transition diagram for the FSM is depicted in Fig. 5 and the complete structure of the state-based neural networks is presented in Fig. 6.

VII. DISCUSSIONS AND CONCLUSIONS

In the above model (see Fig. 6) the FSM block can be model as neural networks. Finite-state machines (formulated as Mealy machines) can be implemented using neuron. This is demonstrated in [37], with the implementation of simple automata such as gates, switches, impulse counters, and simple arithmetic circuits. This fact provides the basis for the connection and equivalence of artificial neural networks and the formal theories of language and computation. In Minsky's own words, "every finite-state machine is equivalent to, and can be simulated by, some neural net". A theorem to support this idea, constructs a recurrent neural net which detect a particular combination of state and input symbol and units which compute outputs. The neural net is shown to exhibit the same behavior as the corresponding Mealy machine [37]. Implementing the FSM as recurrent neural network will in the long run make the system to behave as standard RNN. Furthermore, allowing the FSM so behave as a static machine in which each state is represented with a set of symbol would make the system to have a state (memory) at any time t , but this is limited to binary inputs, it is however, severely limited by the number of states, since state grows as the number of inputs to be represented increases.

We are currently working on the implementation of model to test it potentials. We also believed the model has other potential applications such as voice activity detection (VAD). The neural network model can also be replaced with support vector machine (SVM).

REFERENCES

- [1] R. Sun and C. L. Giles, *Introduction to Sequence Learning, Sequence Learning: Paradigms, Algorithms, and Applications*, Springer, 2001, pp. 1-10.
- [2] P. Baldi *et al.*, "Bidirectional dynamics for protein secondary structure prediction," in *Sequence Learning: Paradigms, Algorithms, and Applications*, R. Sun and C. L. Giles, Eds. Springer, 2001, pp. 80-104.
- [3] M. Sabri, "Recognition of writer independent offline handwritten Numerals using hidden markov models," *Signal Processing*, vol. 88, pp. 844-857, 2008.
- [4] S. Alma'adeed and C. E. Higgins, "Offline recognition of handwritten arabic words using multiple HMM," *Knowledge Based Systems*, vol. 17, pp. 75-79, 2004.
- [5] D. Gulin and H. S. Murat, "Speech recognition with artificial neural networks," *Digital Signal Processing*, vol. 20, 2010.
- [6] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, pp. 91-126, 2001.
- [7] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience Publication, 2001.
- [8] M. A. Dia, "Utilizing knowledge and data driven to enhance Arabic recognition system," *Computer Science and Engineering*, King Fahd University of Petroleum and Minerals: Dhahran, 2012.
- [9] T. P. Mohammad, *Arabic Text Recognition System*, King Fahd University of petroleum and Minerals: Dhahran, 2010.
- [10] C. Mohammed *et al.*, *Character Recognition Systems: A Guide to Students and Practitioners*, Wiley publishers, 2007.
- [11] A. T. A. Taani, "Recognition of online Arabic handwritten characters using structural features," *Language*, vol. 1, pp. 23-37, 2010.
- [12] L. Yann *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.
- [13] C. M. John, *Introduction to Languages and Theory of Computation*, McGraw Hill, 2010.
- [14] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata theory, Languages and Computation*, MA: Addison-Wesley, 1979.
- [15] R. M. Keller, *Computer Science: Abstraction to Implementation*, Harvey Mudd College, 2001.
- [16] A. Aho, *Compiler Principles*, Prentice Hall, 1985.
- [17] N. Chomsky and M. P. Schutzenberger, "The algebraic theory of context-free languages," *Computer Programming and Formal Systems*, E. P. Bradford and D. Hirschberg, Ed. Amsterdam, Netherland, 1963, pp. 118-161.
- [18] H. Bourlard and H. Bengio, "Hidden Markov models and other finite state automata for sequence processing," *Idiap Research Paper*, 2001.
- [19] M. Andrea, "Hidden Markov model applied to data mining," *Dipartimento di Informatica*, Italy, 2006.
- [20] R. R. Lawrence, "A tutorial on hidden Markov model and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, 1989.
- [21] T. Xian, "Hybrid hidden Markov model and artificial neural network for automatic speech recognition," in *Proc. the 2009 Pacific-Asia Conference on Circuits, Communications and Systems*, 2009, pp. 682-685.
- [22] S. Haykin, *Neural Networks and Machine Learning*, Pearson International Edition, 2009.
- [23] S. Masahide, S. Hidehumi, and A. H. Waibel, "Review of TDNN (time delay neural network) architectures for speech recognition," in *Proc. the IEEE Conference on Circuits and System*, 1991, vol. 1, pp. 582-585.
- [24] J. Bromley *et al.*, "Signature verification using a siamese time delay neural network," *Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 669-687, 1993.
- [25] L. Bottou *et al.*, "Speaker independent isolated digit recognition: multilayer perceptron vs dynamic time warping," *Neural Networks*, vol. 3, pp. 453-465, 1990.
- [26] Wiki, *Recurrent Neural Networks*, 2014.
- [27] E. O. Dijk, "Analysis of recurrent neural networks with applications to independent speaker phoneme recognition," *Electrical Department*, University of Twente, Enschede, Netherland, 1999.
- [28] K. Kasper *et al.*, "A fully recurrent neural network for recognition of noisy telephone speech," in *Proc. the IEEE conference on Acoustics, Speech and Signal*, 1995, vol. 5, pp. 3331-3334.
- [29] D. S. Clouse *et al.*, "Time-delay neural networks: Representation and induction of finite state machines," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1065-1070, 1997.
- [30] H. Jaeger, *Echo State Network*, 2007.
- [31] J. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- [32] R. Raul, *Neural Network: As Cybernetic Systems*, 2nd ed., Holk Cruise, 1996.
- [33] S. Hochreiter and J. Schmidhuben, "Long short term memory," *Neural Computations*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [34] A. Graves and J. Schmidhuben, "Offline handwriting recognition with multidimensional recurrent neural networks," *Advances in Neural Network Processing*, Vancouver: MIT Press, vol. 22, 2009.
- [35] S. Fernandez, A. Graves, and J. Schmidhuben, "An application of recurrent neural networks to discriminative keyword spotting," in *Proc. the International Conf. on Artificial Neural Network*, 2007, pp. 220-229.
- [36] H. Kwakernaak and R. Sivan, *Modern Signal and Systems*, New Jersey, USA: Prentice-Hall, 1991.
- [37] L. F. Mikel, "Neural networks: Automata and formal model of computations," *Alacant*, Spain, 2012.



Mohammed Onimisi Yahaya is currently an assistant professor at the College of Computer Science and Engineering Technology, King Fahd University of Petroleum and Minerals, Affiliated Colleges at Hafr Al Batin. He received the PhD degree in computer science and engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in 2014.