Anatolian Journal of Computer Sciences

Volume:5 No:2 2020 © Anatolian Science

pp:71-89 ISSN:2548-1304

AMI: An Auditory Machine Intelligence Algorithm for Predicting Sensory-Like Data

E.N. Osegi[1], V.I. Anireh[2].*

*[*1]National Open University of Nigeria (NOUN), Information Technology, Lagos, Nigeria (nd.osegi@sure-gp.com)*

[2] Rivers State University, Computer Science, Rivers, Nigeria (anireh.ike@ust.edu.ng)

Received Date : Mar. 23, 2020.

Acceptance Date : May 27, 2020.

Published Date : Dec. 1, 2020

Abstract— In this paper, we present the results of our experiments using a new biologically constrained machine intelligence algorithm based on neural processing in the auditory cortex called auditory machine intelligence (AMI). This algorithm is an online learning technique for predicting sensory time series data i.e. data that comes in streams or a sequential order. The AMI algorithm is particularly inspired by the mismatch negativity effect which provides important evidence that the brain learns a statistical structure of the world it senses. We show through a number of experiments with popular benchmarks, how this algorithm may be applied in a real world sense. The results of these experiments have also been compared with two very popular techniques that have been used for time series predictions and are very encouraging.

Keywords : *Auditory processing, biological machine intelligence, predictions, sensory data, time series.*

1. BACKGROUND

Sequence or time-series prediction of real world sensory data plays a very vital role in modern day society as the ever increasing need to deliver more useful products increases and the need to guarantee minimal or zero service down-times with effective product distributions and higher sales/profits rises on a continual basis. For instance, some important sensory data prediction estimates required by service delivery agents include the times of day a potential customer will visit a retail store, the number of times a particular product or commodity is demanded and the amount traded, the fluctuating demand for electrical load at power station centres etc. Some other more specific examples include forecasting the opening and closing stock prices in a stock exchange, forecasting weather events in order to determine flight schedules or business trips and determining the required electrical power generation schedule from electrical power demand predictions.

More recently, there has been a surge in interest on the use of artificial intelligence (AI) techniques and algorithms for making reliable prediction estimates. However, the problem of developing an AI algorithm that is general enough and that can solve a variety of tasks has been an obvious challenge to AI experts across the globe particularly as it pertains sequence (time-series) predictions of the aforementioned examples. In particular, AI algorithms that can make adaptive predictions or decode possible (future) state(s) of an incoming sensory signal continually are indeed of particular interest in both academic and industry setting. However, it is not always possible for an algorithm to perform better than others in all the assigned tasks. One possible explanation for this situation is derived from the 'No Free Lunch Theorem' or simply NFLT which generalizes the problem and hypothesizes that it is not possible for any algorithm to do better than another in all tasks (Wolpert and Macready, 1997).

Classical Artificial Neural Networks (ANN), also known as universal function approximators have been very successful in the prediction of a number of time series related data. Some of such popular ANN or neural machine learning techniques include the Long Short-Term Memory (LSTM) developed in (Hochreiter and Schmidhuber, 1997), the Group Method of Data Handling (GMDH) developed in (Ivakhnenko, 1968), and the Online-Sequential Extreme Learning Machines (OS-ELM) developed in (Liang et al., 2006). These algorithms have indeed been successfully applied in many tasks and due to space restrictions we do not list them all here. However, their inherent complexity particularly the case of excessive hyper-parameter tuning and the need for very large data for training makes such neural networks incompatible prediction algorithms in real time scenarios.

Recently, a class of artificial neural networks (ANNs) called the Hierarchical Temporal Memory (HTM) has been proposed as a candidate solution as par developing AI algorithms that are general enough to be used in a variety of real world prediction tasks and that work in accordance to the principles of the brain (Hawkins et al., 2010; Hawkins et al., 2016). However, HTM still faces the challenges of biological constrain and unnecessary complexity of the temporal part always requiring the need for a time stamp attribute in most sequence level data mining tasks (Struye and Latré, 2019); thus, its reliability in certain applications may be in doubt. Indeed, the HTM-like algorithm may require some exhaustive sort of parameter tuning to work in certain time series related prediction problems.

For machine intelligence algorithms that must adapt to changing statistics in a timely and more reliable manner, a reduction in the inherent complexities in their temporal processing and parameterization step cannot be over-emphasized.

In this paper, we propose yet another online learning machine intelligence algorithm, the Auditory Machine Intelligence (AMI), that is though more biologically constrained, but is less complex than HTM obviating the need for a time stamp attribute and does perform well in a number of time series prediction tasks. Following the idea and findings about the Mismatch Negativity Effect (MMN) we have developed an algorithm that can learn to give more precise predictions in a deterministic manner through time. The emphasis of this research is not necessarily on bettering the state-of-the-art neural algorithms/techniques, but on developing a simple but yet effective continual machine learning neural technique that is general enough to be applied to various problems in different application domains – software and on embedded hardware prediction problems inclusive.

Our research paper is organized as follows: Section 2 will present briefly previous work in the state-of-the-art highlighting the key algorithms that have been used for sequence prediction of data. Section 3 describes our proposed methodology. Section 4 will present the experiments; comparisons will be made with two state-of-the-art sequence learning neural networks that have been used more recently for time series prediction: the HTM which is a biological based neural network that is trained using Hebbian learning rules and the LSTM – a more traditional (mathematical) neural network that is trained using the very popular back-propagation (gradient descent) algorithm. Specific comparisons will also be made with the HTM using real-world sensory datasets obtained from the Numenta Anomaly Benchmark. In Section 5 we present our conclusions and discuss some shortcomings of our proposed technique and how the AMI may be improved upon.

1.1. Statement Of Problem

Time series prediction is very challenging and algorithms that have been developed for sequence learning tasks such as the Online-Sequential Extreme Learning Machines (Liang et al., 2006), the Group Method of Data Handling (Ivakhnenko, 1998), the Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and the Hierarchical Temporal Memory which is based on the Cortical Learning Algorithms (HTM-CLA) developed in (Hawkins et al., 2016) have been useful but can be very expensive/time consuming to implement in real world business or industrial application software. It is also particularly very challenging to develop such algorithms on embedded hardware and development time is increased when building on cloud computing frameworks. Thus, an algorithm that is powerful but possesses simplicity is highly desired particularly for predictive applications that require rapid deployment.

1.2. Research Objective

The primary objective of this research is to present a new neuro-biological auditory inspired but constrained approach that is simple and useful for the prediction of time series data or any type of sensory data that is characterized by sequential changes through time. In order to attain this first objective, we leverage on existing neuro-biological principles and theories related to the studies of the mismatch negativity effect (MMN) and a brain region called A1 located in the auditory cortex.

It is also the objective of this research to compare our proposed (AMI) technique with a traditional sequence learning neural and statistical prediction model using several open source and real time series benchmark datasets.

2. Previous Work

Deep neural networks such as the Long Short-Term Memory (Hochreiter & Schmidhuber, 1997; Gers et al., 2000) and the Group Method of Data Handling (Ivakhnenko, 1998) emerged as possible candidates for predictive classification tasks. This class of machine learning neural algorithms can operate in unsupervised/semi-supervised levels and have been particularly useful in solving a variety of real world challenging problems including but not limited to speech recognition (Graves et al., 2013), image recognition (Sutskever & Hinton, 2007), sentiment analysis and natural language processing (Socher et al, 2011a, Socher et al, 2011b). One primary reason for the success of deep neural networks is their ability to learn multiple layers of representation (Hinton et al, 2007). Notwithstanding these promising benefits, the deep neural network models still require extensive hand tuning making it very difficult to implement in real world tasks (Cui et al., 2016b). Another drawback is their computational cost and over reliance on very big data which in most cases is not readily available, too expensive to process and in some instances may not be necessary particularly for real-time applications that use sensor/actuators.

HTM, an online learning machine intelligence and an emerging sequence learning neural technique has been particularly very useful for analyzing streaming sensory data i.e. continual data with a changing statistic through time. This includes short-term load forecasting (Osegi et al., 2018) and prediction of taxi passenger data (Cui et al, 2016a; Cui et al, 2016b). HTM is also very useful as an anomaly detector and have been applied to a variety of tasks including but not limited to weld flaw detection (Rodriguez-Cobo et al, 2013) and anomaly detection in streaming data (Ahmad et al, 2017).

In the subsequent section, we present the details of a biologically constrained but less complex machine intelligence neural technique.

3. AUDITORY MACHINE INTELLIGENCE

Auditory Machine Intelligence (AMI) is a type of neural machine intelligence technique that fundamentally uses time dependent deterministic processing for time series prediction. The primary purpose of the AMI technique was to overcome the limitations in the temporal learning phase of the existing Hierarchical Temporal Memory (HTM) neural techniques developed in (Cui et al., 2016a, Cui et al., 2016b & Cui et al., 2017). In an AMI, the concept of mismatch negativity effect (MMN) and intelligent processing in mammalian auditory cortex are exploited to develop a precise and reliable algorithm that can give more precise predictions in a timely and deterministic way. AMI was introduced earlier in (Osegi & Anireh, 2016) bearing the name of "Deviant Learning Algorithm". Some recent applications can be found in (Osegi et al., 2018) where it was used for time series prediction of Internet-of-Things (IoT) data, in the prediction of critical clearing time of a resonant fault limit electrical protection system (Wokoma & Osegi, 2019) and in a very recent application, the dynamic load prediction for smart grid applications (Osegi et al., 2019).

AMI exploits the idea and findings about the mismatch negativity effect (MMN) and intelligent processing in mature mammalian auditory cortex (called A1), to build an algorithm that can give more precise predictions in a timely and more reliable manner. The MMN is a differential neuronal response to a repetition of stimulus presentations and an oddball or deviant stimulus signal (Takaura & Fuji, 2016). Indeed, MMN has been shown to exhibit very useful properties for high-level cognitive operations (Näätänen et al., 2007).

Recent experiments on macaque monkeys using electro-corticography in the context of the roving odd-ball has shown the MMA (corresponding MMN effect in animals) to exhibit Specific-Stimulus Adaptation (SSA) properties including cortical response to deviant stimulus (odd-balls) over a wide area of cortical regions (Takaura & Fujii, 2016). In (Sollini et al., 2018), it has also been shown via neuronal simulations that functional reorganization of the receptive fields in A1, will occur when subjected to appropriate frequency fine-tuning and exposure of the cortical front (ON/OFF) receptive fields to sound waves. Thus, in a developing auditory cortex, the ON/OFF receptive fields sensitivity to sensory signals (frequency modulated tone sweeps) can be explained by using a simple universally accepted hebbian rule. A visual concept of this frequency tuned learning for functional reorganization is as shown in Fig. 1; in the visual, the large white blob represent a neuron, excitatory and inhibitory ON synaptic weight inputs are indicated as green, open and filled blobs while excitatory and inhibitory OFF inputs are indicated as red, open and filled blobs. As illustrated in Fig. 1, after hebbian learning operation, a divergence in synaptic weight inputs (indicated by the thick dark connected arrows) to the processed or simulated neuron (now grey) occurs.

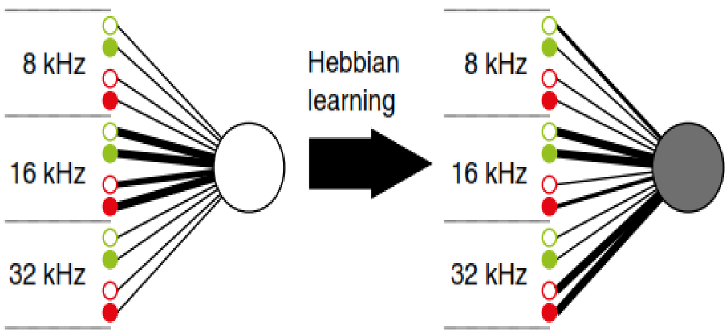

Fig. 1. A Visual of a freqency-tuned hebbian learning concept in A1 (Sollini et al., 2018).

In particular, at high frequencies, the divergence should be larger in magnitude while at lower frequencies this should be lower. Thus, learning may be viewed as a set of adaptive synaptic connecting links where the rate or level of connection is frequency-dependent.

The following sub-section (sub-section 3.1) presents and describes the details of the MMN effect theory and highlights two important application theories used in the AMI while in sub-section 3.2, a mathematization of the processes that describe the principle of operation of the core AMI technique is also presented.

3.1. Concept of the Mismatch Negativity Effect

The Mismatch Negativity Effect (MMN) typically represents a differential neuronal response to a repetition of stimulus presentations and a consequent oddball or deviant stimulus signal (Takaura and Fuji, 2016). The MMN exhibits very important high level cognitive processes including such key processes as grammar and semantic meaning higher perception and cognition functions such as in speech and music etc (Näätänen et al., 2007). In recent electro-corticography experiments on macaque monkeys, it has also been shown that the Mismatch Activity (MMA) – the analogue of the MMN effect in animals exhibits Specific-Stimulus Adaptation (SSA) properties including cortical response to deviant stimulus signals or odd balls over wide cortical regions (Takaura and Fuji, 2016).

In subsections 3.1.1-3.1.3, the theoretical models related to the MMN effect including the Change Detection (CD) and Model Adjustment (MA) theories are presented while subsection 3.1.4 details the AMI neuronal structure.

3.1.1. Theoretical Models of the Mismatch Negativity Effect

Lieder et al (2013a), proposed a theoretical framework including five theoretical models for describing the statistical structure and importance of the Mismatch Negativity (MMN) effect in auditory stimulation. These models are categorized into two frameworks:

- The Phenomenological (PM) Framework - which includes the Change Detection (CD) and Neural Adaptation (NA) theories, and,

- The Free Energy Principle (FEP) which includes the Prediction Error (PE), Novelty Detection (ND) and the Model Adjustment (MA) theories.

The PM and FEP leads to the derivation of 13 computational models (or response functions) of a generalized state space model. The response functions can be found in (Lieder et al 2013a, table 2) and are based on the model expressions in Eq. (1) and Eq. (2):

$$x_{t+1} = f\left(x_t, u_t; \theta_p\right) \tag{1}$$

$$y_t = g\left(x_t, u_t; \theta_p\right)\beta + \varepsilon_t \tag{2}$$

where,

x_t = a generative non-observable internal state

f = an evolution function describing x,

$y_t =$ a mapping of all x to a sensory input, u,

g = a response function describing y,

$\theta_p =$ the state space parameters.

$\varepsilon = \mathrm{N}\left(0, \sigma^2\right)$

$\sigma =$ a representation of the variance of the medial geniculate nucleus.

Using Eq. (1) and Eq. (2), a generative function can be formulated as:

$$M = \left\{f, g, p(\theta)\right\} \tag{3}$$

where *p(θ)* represents a prior density function.

The evolution function, response function in conjunction with the prior density function describes the core features of a generative model based on the MMN effect. Amongst the theories formalized in (Lieder et al., 2013a), the CD theory features more prominently in the AMI and will be described subsequently in the subsequent sub-section; this will then be followed by the Model Adjustment (MA) theory.

3.1.2. Change Detection in the AMI

As mentioned earlier in section 3.1.1, the CD theory prominently features in the AMI as most sensor based time series problems are of the univariate type. This theory quantitatively defines the following MMN mismatch predictions/operations expressed as response functions (Lieder et al., 2013a):

- An MMN indexes only when a change occurs or not. This is denoted by the response function g1.
- An MMN indexes the unsigned or absolute value of the change in a physical property of a sensory input signal. This is denoted by the response function g2.
- An MMN indexes the signed value of the change in a physical property of a sensory input. This is denoted by the response function g3.

The response functions modelling these predictions are given in Eq. (4) to Eq. (6) and are all mimicked by the AMI in software.

$$g_1 = \begin{cases} 1, & if\ u_t \neq x_t \\ 0, & otherwise \end{cases} \tag{4}$$

$$g_2 = \left|u_t - x_t\right| \tag{5}$$

$$g_3 = u_t - x_t \tag{6}$$

Note that, x_t and u_t are the non-observable generative internal state and sensory input respectively.

3.1.3. Model Adjustment in the AMI

In the AMI, performing level-2 mismatch operations requires certain adjustments to be made during predictions. The Model Adjustment (MA) theory provides support for this operation and is indexed programmatically by adjusting a sequence length:

$$g_{13} = \begin{cases} \left| x_{in_{(z+1)}} - x_{store_{(z)}} \right| \leq \rho_{2}, \\ 0 \quad otherwise \end{cases} \tag{7}$$

where,
ρ_2 = a permanence threshold value
x_{in} = is the input observation and
x_{store} = sparse representation of x_{in}.

Consider a group of observation sequences say S_m; then the probability that a sequence say S_o, will be correctly predicted at the next time step may be approximated as:

$$\mathrm{P}\left(S^{1}_{o(t)} \middle| S_{o(t-1)} \right) = \frac{\left(\sum \left| S_{o_{(t-1)}} - S_{p_n} \right| \geq T_d \right)}{N_z}, \quad if \; \exists S_o \ni S_o \subset S_m \tag{8}$$

where,
T_d = a deviant overlap threshold value,
S_p = the sparse generated predictions,
$S'_{o(t-1)}$ = the future prediction of S_o and
N_z = size of S_m.

3.1.4. AMI Neuronal Structure

The AMI primary formula as described in Section 3.1 and as outlined in Algorithm 1-2 (sub-sections 3.1 and 3.2) is as captured in the neural schematic of Fig.2. In this diagram, mathematical formulas are described by the operator sign while the functional modules and a trigger block describe key functional routines and time series feature control initializations used in the AMI. The Binary Encoder and Binary-to-Integer Transformer modules are used to convert the inputs (X_t) from a multivariate to a univariate time series. By default, the CD mismatch processing is enabled and a trigger control is set to 0. When a transition is needed from a univariate to a multivariate time series processing, the trigger is enabled and the MA processing of X_t is called upon; otherwise it remains disabled and the CD processing of X_t continues.

One important feature of this neural architecture is the availability of a dual-prediction output. This feature accounts for the two prominent MMN properties – the CD property which indexes a response function as a difference between sensory observation and an internal generative state and the MA property which permits adjustments to be made on the sequence length of a set of sensory observations in accordance to a threshold limitation (Leider et al., 2013a). This second property facilitates level-2 mismatch operations to be performed by the AMI. In most instances of the AMI predictions, a univariate time series persists and a CD operation is assumed. Where the time series is of a multivariate form, the MA function is called into action through the aforementioned AMI's special decoding circuit.

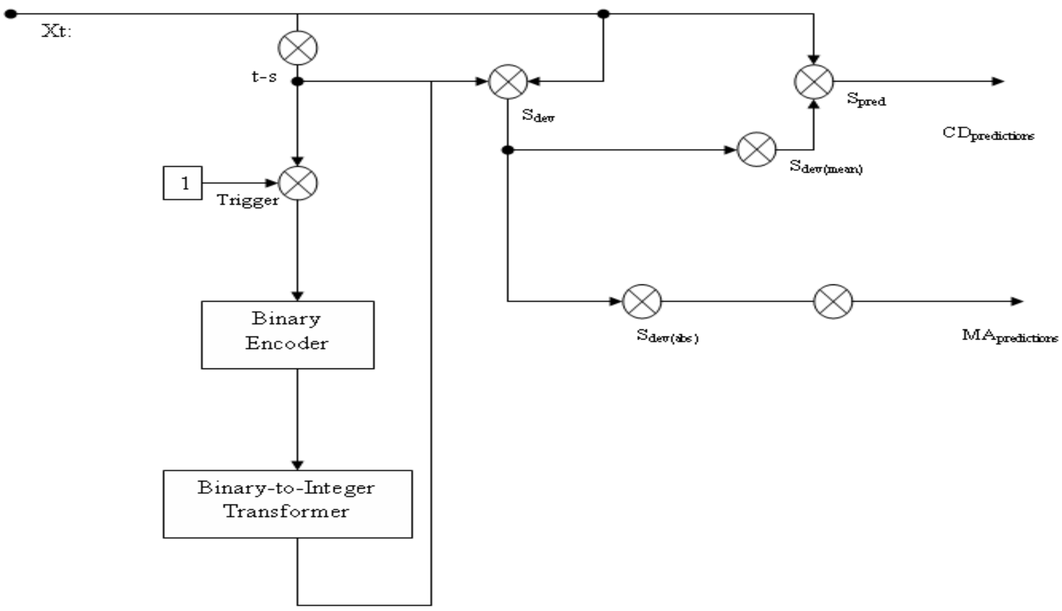

Fig.2. AMI Neural Structure.

3.2. AMI Operational Principle

The AMI principle is based on the MMN theory (Takaura & Fujii, 2016; Näätänen et al., 1978, 2007; Lieder et al., 2013a, 2013b), and on the theory of functional reorganization in mammalian auditory cortex, A1 (Sollini et al., 2018). In the context of MMN, the change detection theory features more prominently. Furthermore, the concept of functional reorganization is exhibited by a frequency-tuned hebbian-like learning mechanism.

The prediction operations in the AMI basically occur in two-phases (Osegi et al., 2018):

- Phase-1 (low-level) prediction of an observed time series value in the current time step based on a history of time series data values several time steps back. These data values may be sparse and they correspond to the evoked potentials originally observed by Näätänen et al (1978) as the "odd-ball" phenomenon.
- Phase-2 (top-level) prediction for performing predictions several time steps in advance.

The details of these phases of predictions are presented in the extended sub-sections that follow (sub-sections 3.2.1 and 3.2.2).

3.2.1. Phase-1 Predictions in the AMI

In the AMI, a Phase-1 prediction is used for making predictions one-step ahead. This is basically done using a single learning formula and random fine-tuning is not necessary for learning purposes. Specifically, in phase-1 predictions, the AMI learns a sequence of data points (values) automatically/temporally in an adaptive manner such that a mean deviant point is computed as in Eq.(8):

$$S_{dev(mean)} = \frac{\left(\left(\frac{\sum[S_{dev}]}{(n-1)}\right) + S_{deviant}\right) - 2}{n+1} \tag{8}$$

where,

n = number of data points in a temporal sequence

$S_{deviant}$ = the *(n-1)th* value of the temporal sequence

S_{dev} = the difference between $S_{deviant}$ and S_{stars}

S_{stars} = the *(n-2)th* values of the temporal sequence

S^* = sparse set of input sequences

In order to make a prediction with the AMI, the considered deviant is added to the mean deviant as:

$$S_{pred} = S_{deviant} + S_{dev(mean)} \tag{9}$$

where,

$$S_{deviant} = S_n^* - 1 \tag{10}$$

$$S_{stars} = S_n^* - 2 \tag{11}$$

The AMI algorithm is as described in Algorithm 1.

Algorithm 1. AMI Processing Algorithm

A. 1: Initialize S_{pred}, as prediction parameter, S_{stars}, as input sequences (standards) State, $S_{dev(mean)}$ as deviant mean, *j* as iteration counter.
B. 2: *for* all $s \in s.S_{stars}$, && $j > 1$, do
C. 3: Compute $S_{deviant}$ and S_{stars} using equations Eq.(10) and Eq.(11)
D. 4: $S_{dev} \leftarrow \| S_{deviant} - S_{stars} \|$ // deviations from standards
E. 5: Compute $S_{dev(mean)}$ using equations Eq.(8)
F. 6: Compute S_{pred} using equations Eq.(8) and Eq.(10)
G. 7: Update $S_{dev(mean)}$ using Algorithm 2
H. 8: *end for*

The AMI algorithm uses the concept of hebbian learning and is described in the following way:

If the current prediction error of the AMI neuron is greater than or lower than zero, its prediction value is reinforced by decreasing or increasing its deviant weight value by the absolute prediction error difference at the current time step and incrementing its recognition threshold by a small factor; otherwise a zero or negligible positive reinforcement is used by adding a very small value (deviant-laplacian correction). In instances where exact matches occur, a small laplacian correction value (typically in small fractions of about a hundredth), is also used for deviant weight updates. The learning rule is described succinctly as in Algorithm 2.

Algorithm 2. AMI Learning Algorithm

İ. 1: Initialize S_{pred}, as prediction parameter, S_{stars}, as input sequences (standards) State, $S_{dev(mean)}$ as deviant mean, $S_{diff(1)}$ as difference between S_{pred}, $S_{deviant}+1$ and $S_{diff(2)}$ as difference between $S_{dev(mean)}$ and $|S_{diff(1)}|$, l_p as correction factor or bias, T_h as recognition threshold.
J. 2: *for* all $s \in s.S_{stars}$ do
K. 3: if $S_{diff(2)} > 0$
L. 4: $S_{dev(mean)} \leftarrow S_{dev(mean)} - |S_{diff(1)}|$ // Weaken deviant mean by a factor, $|S_{diff(1)}|$
M. 5: $T_h = \min(T_h + 0.01,\ 1)$ // Positive recognition threshold reinforcement
N. 6: elseif $S_{diff(2)} < 0$
O. 7: $S_{dev(mean)} \leftarrow S_{dev(mean)} + |S_{diff(1)}|$ // Reinforce deviant mean by a factor, $|S_{diff(1)}|$
P. 8: else
Q. 9: $S_{dev(mean)} \leftarrow S_{dev(mean)} + l_p$
R. 10: $T_h = T_h + 0.01$ // recognition threshold bias
S. 11: end if
T. 12: *end for*

3.2.2. Modelling State Transitions - Phase-2 Predictions in the AMI

In a typical time series prediction, there are several possibilities that may exist when the context prediction representation is defined by the time steps. For instance, if we take the current time step as, t_2, then at the minimum look-ahead/look-back case, the past and future time steps are t_1 and t_3 respectively. However, our goal is to predict the time series value, V_{t3}, at the future time step using the time series values at the current and past time steps namely, V_{t2} and V_{t1}. The possible conditional encoded state relations at these time steps are detected in Algorithm 3:

Algorithm 3. Encoding state relations in the AMI

$$
\begin{aligned}
&If \quad (V_{t2} > V_{t1}) \\
&\ encode\_state = 3 \\
&else, \\
&If \quad (V_{t2} \equiv V_{t1}) \\
&\ encode\_state = 2 \\
&else, \\
&If \quad (V_{t2} < V_{t1}) \\
&\ encode\_state = 1
\end{aligned}
$$

These states are then used to update the forecast horizons using the AMI Phase-1 predictions and deviant means as in the decoder technique in Algorithm 4:

Algorithm 4. Decoding Phase-2 Predictions in the AMI

$$
\begin{aligned}
&If \ \ (encode\_state \equiv 3) \\
&\ S_{pred(i)} = S_{pred(i-1)} + S_{dev(phase\ 1)} \\
&else, \\
&If \ \ (encode\_state \equiv 2) \\
&\ S_{pred(i)} = S_{pred(i-1)} \\
&else, \\
&If \ \ (encode\_state \equiv 1) \\
&\ S_{pred(i)} = S_{pred(i-1)} - S_{pred(phase\ 1)}
\end{aligned}
$$

Just as in Phase-1 predictions, the CD theory applies to Phase-2 predictions as well; in this regard, the deviant-phase parameter is computed using Eq,(8).

4. EXPERIMENTS AND RESULTS

4.1. Experiment Details

The experiments presented in this research are performed in several parts:

In the first part (section 4.2), a single/double character learning problem is presented; this experiment is used as a graphic demonstration of the symbolic prediction and representational capability of the AMI neural technique in making continual predictions.

In the second part (section 4.3), the experiments using some popular and recent time series data are presented. Where the data is of the univariate type, it is fed directly to the AMI and the predictions decoded through time using the AMI Change Detection (CD) principle (see Section 3, sub-section 3.1.2).

In the case when the data is of the multivariate type, we use a binarization routine to encode the inputs into a sparse representation, and then train the inputs online using the Model-Adjustment (MA) rule described earlier in Section 3 (sub-section 3.1.3). The idea in this approach is to demonstrate the model switching high-level transfer learning capability of the AMI technique. Earlier versions of the AMI (Osegi et al., 2018) can only handle univariate input. We believe that by including this feature, developers can leverage on the full capacity of the AMI neural processing functions to give a more generalized predictive representation.

4.1.1. Benchmarked Data of the State-of-the-art

The experimental studies including simulation results using 5 standard benchmark data sets for time series problems and 4 Numenta Anomaly Benchmark (NAB) datasets for real time streaming data which have recently been proposed in (Lavin and Ahmad, 2015) are presented; these experiments are provided in sections 4.3 and 4.4 respectively. The experimental proof is intended to show that our algorithm is scalable in a variety of tasks - both for standard time series benchmark data and streaming benchmark data. The feature size and details of the standard time series dataset are presented in Table 1 while that of the NAB datasets are presented in Table 2.

Table 1 Standard datasets. Datasets 1-4 (Moritz et al., 2015); dataset 5 (Yöntem et al., 2019)

Id	Benchmark	Description	Number of Exemplars	Number of attributes
1	airpass	Monthly total international airline passengers from 01/1960 - 12/1971	144	1
2	beersales	Monthly beer sales in millions of barrels, 01/1975 - 12/1990	192	1
3	SP	Quarterly S&P Composite Stock Index, 1936Q1 - 1977Q4	521	1
4	google	Daily returns of the google stock from 08/20/04 - 09/13/06	168	1
5	divorce	marital perceptions of divorced and happily married persons	170	54

Table 2 Numenta Anomaly Benchmark (NAB) data (Lavin and Ahmad, 2015)

id	Benchmark	Description	Number of Exemplars	Number of attributes
1	speed_7578	speed obtained from specific sensors	1127	1
2	nyc_taxi	New York city taxi data	10320	1
3	rogue_agent_key_hold	timing anomalies that occur when several users press the keys of a computer	1182	1
4	hot_gym	Energy consumption data from an Australian gymnasium	4391	1

4.1.2. State-of-the-art machine learning neural techniques

In order to compare our proposed technique, we have considered two state-of-the-art machine learning neural techniques. These techniques include the Long Short-Term Memory, a recurrent type neural network that has gained popularity in recent times and the Hierarchical Temporal Memory (HTM) which is based on the Cortical Learning Algorithms. The parameters of these two techniques are presented in Appendix A.

4.2. The Single-to-Double Character Learning Problem Experiments

These experiments are conducted in order to gain insight into the pattern prediction behaviour of the proposed AMI technique. Sample character sequences for the single and double character problem under study as presented to the AMI program is given in Table 3. These sequences represent the characters of the English alphabet in their capitalized form. The sequences are presented character by character, one step at a time and the task is for the AMI to tell in advance what the next character in the sequence will be. For the single character problem, the sequences are an ordered replicated representation of the alphabets, A, B and C as "A, B, C, A, B, C… A, B, C" in that order for the first 73 values and these values are presented to the AMI one step at a time. For the next 23 symbols, a steady symbol representation, "A", is presented to the AMI one step at a time.

For the double character problem, the sequences are a mixture of some of the English alphabets in bi-variate form and with most patterns of the form "AB".

Table 3 Sample sequences for the AMI character learning problem

Single Character Sequence	**Double Character Sequence**
A	AB
B	CD
C	EF
A	GH
B	IJ
C	KL
A	MN
B	AB
C	AB
A	AB
A	AB
A	AB
A	AB
A	AB
A	AB

4.2.1 Single Character Learning

The sample numerical results of simulations for single character problem showing the continuous learning performance of AMI are provided in Table 4. The predictions are performed one-step ahead and in a continuous manner.

The results show that for the single-character problem, the learning impact of AMI on the data time series is improved with time; a continual Mean Absolute Percentage Error (MAPE) plot (see Fig. 3) also clearly shows that the AMI will improve its prediction error minimization response as the patterns starts becoming more regular.

Table 4. Predictions of AMI for single character learning problem

Predicted Symbol	Actual Symbol	Predicted Symbol	Actual Symbol	Predicted Symbol	Actual Symbol	Predicted Symbol	Actual Symbol
@	B	A	C	G	A	B	A
A	C	I	A	?	B	A	A
_	A	?	B	A	C	A	A
@	B	A	C	G	A	A	A
A	C	H	A	@	B	A	A
U	A	?	B	A	C	B	A
?	B	A	C	G	A	A	A
A	C	H	A	@	B	B	A
Q	A	@	B	A	C	A	A
@	B	A	C	F	A	B	A
A	C	H	A	@	B	A	A
N	A	@	B	A	C	A	A
@	B	A	C	F	A		
A	C	H	A	@	B		
L	A	@	B	A	C		
@	B	A	C	F	A		
A	C	G	A	A	A		
K	A	@	B	B	A		
@	B	A	C	B	A		
A	C	G	A	B	A		
J	A	?	B	A	A		
?	B	A	C	B	A		
A	C	G	A	A	A		
J	A	@	B	A	A		
@	B	A	C	B	A		
A	C	G	A	B	A		
I	A	?	B	A	A		
@	B	A	C	B	A		

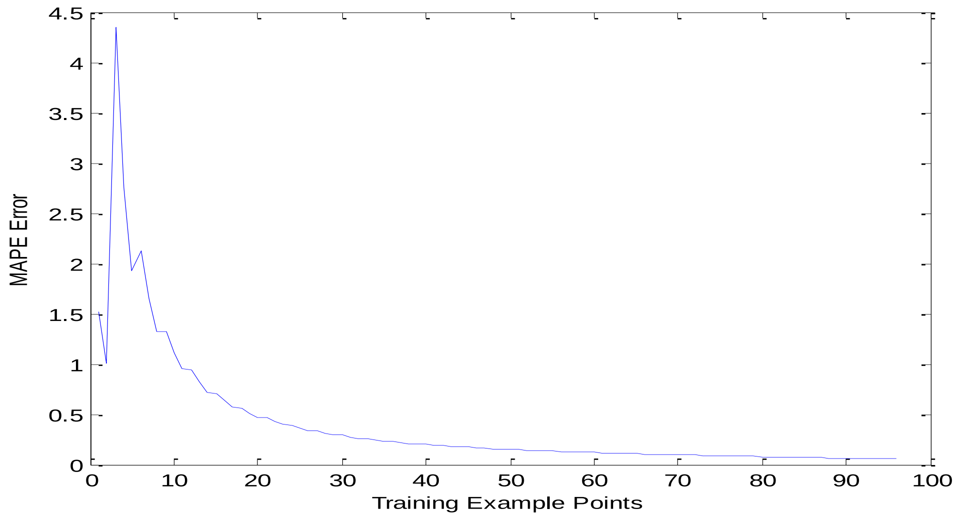

Fig.3. Continual MAPE response for the single-character problem

4.2.2 Double Character Learning

The sample numerical results of simulations for double character problem showing the performance of AMI using threshold recognition factors, T_h, from 0.1 to a point where the T_h update just equals 1 are provided in this experiment. This fine tuning step is done at increments of 0.1 and is equivalent to setting the linear spacing between 0.1 and 1.0 at most 10 point of intervals. The first setting for T_h (T_h, = 0.1) and the last setting (T_h, = 0.9) are reported in Tables 5 and 6 respectively; no noticeable improvements over the first setting were observed for the other different settings of T_h.

Table 5 Predictions of AMI for double character learning problem; threshold setting T_h, is at 0.1.

Predicted Symbol	**Actual Symbol**	**Predicted Symbol**	**Actual Symbol**
AA	IJ	CD	KL
BC	KL	CD	MN
CD	MN	CD	AB
DE	AB	CD	AB
EF	AB	CD	AB
FG	AB	DE	AB
GH	AB	DE	AB
FG	AB	DE	AB
EF	AB	CD	AB
EF	AB	CD	AB
DE	AB	CD	AB
DE	AB	CD	AB
DE	AB	CD	AB
DE	AB	CD	AB
CD	AB	CD	AB
CD	AB		
CD	CD		
CD	EF		
CD	GH		
CD	IJ		

The numerical results obtained when the precision of T_h is increased to two places of decimals are also presented. In this step the equivalent linear space setting is at most 100 point of intervals and the simulation stopping criteria is as in the aforementioned paragraph. Due to space restrictions we restrict the reported thresholds to two best settings of T_h, 0.84 and 0.92 as depicted in Table 7 and Table 8 respectively.

The results show that for the double-character problem, the prediction learning impact of AMI on the data time series is not noticeable at a low precision and when increasing the recognition thresholds to a factor of 0.9 it improves. However, by increasing the precision to two places of decimals, there is a noticeable impact of increasing thresholds. In particular, it was found that a threshold between 0.84 and 0.92 gives a good matching prediction space; however, there may be degrading matching prediction performance as the threshold setting tends to 0 or 1.

Table 6 Predictions of AMI for double character learning problem; threshold setting T_h, is at 0.9

Predicted Symbol	**Actual Symbol**	**Predicted Symbol**	**Actual Symbol**
AA	IJ	AB	KL
BC	KL	JK	MN
CD	MN	IJ	AB
GG	AB	IJ	AB
HI	AB	IJ	AB
IJ	AB	AB	AB
CD	AB	AB	AB
BC	AB	AB	AB
BC	AB	AB	AB
BC	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB		
AB	CD		
AB	EF		
AB	GH		

Table 7 Predictions of AMI for double character learning problem; threshold setting T_h, is at 0.84

Predicted Symbol	**Actual Symbol**	**Predicted Symbol**	**Actual Symbol**
AA	IJ	GH	KL
CC	KL	GH	MN
DE	MN	GH	AB
EF	AB	GH	AB
FG	AB	GH	AB
GH	AB	AB	AB
AA	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB	AB	AB
AB	AB		
AB	CD		
AB	EF		
AB	GH		
HI	IJ		

Table 8 Predictions of AMI for double character learning problem; threshold setting T_h, is at increments of 0.92

Predicted Symbol	Actual Symbol	Predicted Symbol	Actual Symbol
AA	IJ	AB	KL
BC	KL	AB	MN
CD	MN	KL	AB
GG	AB	JK	AB
II	AB	JK	AB
JK	AB	BC	AB
DE	AB	BC	AB
CD	AB	BC	AB
CD	AB	BC	AB
BC	AB	BC	AB
BC	AB	AB	AB
BC	AB	AB	AB
BC	AB	AB	AB
BC	AB	AB	AB
BC	AB	AB	AB
AB	AB		
AB	CD		
AB	EF		
AB	GH		
AB	IJ		

4.3. Experimental details and results using standard benchmarks

The experiments are conducted to determine the comparative performance of the proposed AMI technique with LSTM and HTM techniques. The experimental results are compared on the basis of their cross-entropy error loss as this is well applicable to the machine learning community (Goodfellow et al., 2016). Simulations are based on the parameters set out in Appendix A and are presented in Table 9; in the AMI prediction experiment with divorce dataset, a threshold setting of 0.84 is used while in the LSTM model a specific hyperparameter – the Maximum Epoch size is fine tuned to examine its influence on the prediction error loss. Furthermore, the results using the AMI and HTM are reported in terms of the mean of the computed cross-entropies since they make their predictions in a continual manner. As can be seen from the results and based on the default system parameters, the AMI technique clearly outperforms the LSTM technique in all tasks. For the comparative performance with the HTM, the AMI performs better in all but the Google and Divorce datasets where the performance cross-entropy scores are similar.

In order to investigate the feature of hyper-parameter tuning in the LSTM and its superiority over AMI, we increased the number of epochs from the default value to 500, 1000, and 1500. The results are as provided in Table 10.

The result for the Airpass dataset (Table 10) goes to show that hyper-parameter tuning gives the LSTM technique an inherent advantage as its loss reduces with a graded increase in training epochs. In particular, at an epoch of 1000 and 1500 units, the LSTM will outperform the AMI or HTM techniques. However, as can be observed in Table 10, the AMI outperformed the LSTM for three of the standard datasets (Beersales, SP and Google datasets). For the Divorce dataset, the LSTM fared better than the AMI. This goes to show that the underlying structure of the data is important for sequence-wise predictions. The failure of the LSTM to perform better than the AMI for the aforementioned three datasets may be attributed to the lack of a combined trend and seasonality in the presented datasets (see Moritz et al., 2015). Thus, trend or seasonal information is very vital for sequence-to-sequence predictors like LSTM.

Table 9 Cross-entropy loss scores (ce-loss) for the AMI, LSTM and HTM Techniques using the various standard benchmarks

Benchmark Dataset	**AMI**$_{\text{ce-loss(mean)}}$	**LSTM**$_{\text{ce-loss}}$	**HTM**$_{\text{ce-loss(mean)}}$
Airpass	0.0200	0.7947	0.0300
Beersales	0.0100	0.1985	0.0200
SP	0.0100	0.3045	0.0200
Google	0.0100	0.2235	0.0100
Divorce	0.0100	0.5062	0.0100

Table 10 Cross-entropy loss scores-epochs (ce-loss-epochs) for the LSTM with increasing epochs

Benchmark Dataset	**LSTM**$_{\text{ce-loss500}}$	**LSTM**$_{\text{ce-loss1000}}$	**LSTM**$_{\text{ce-loss1500}}$
Airpass	0.1471	0.0061	0.0139
Beersales	0.1813	0.0659	0.0186
SP	0.2280	0.1715	0.0666
Google	0.2228	0.0449	0.0479
Divorce	0.0039	0.0055	0.0031

4.4. Experimental details and results using NAB data

The experiments are conducted to determine the comparative performance of the proposed AMI technique with HTM technique. The essence of this experiment is to validate the robustness of both neuro-biological techniques in prediction tasks of real world sensory data examples. Since the LSTM is a traditional (classical) neural technique it is not considered in this experiment.

Simulation results are presented in Table 11. The results show the superiority of our proposed algorithm over the HTM technique though both techniques did fairly well in all the presented NAB datasets.

Table 11 Cross-entropy loss scores (ce-loss) for the AMI and HTM Techniques using the NAB datasets

NAB Dataset	**AMI**$_{\text{ce-loss(mean)}}$	**HTM**$_{\text{ce-loss(mean)}}$
speed_7578	0.003100	0.010000
nyc_taxi	0.000540	0.000758
rogue_agent_key_hold	0.001390	0.002778
hot_gym	0.000935	0.001638

5. CONCLUDING REMARKS AND FUTURE WORK

In this paper, a novel machine intelligence algorithm of reduced complexity called the Auditory Machine Intelligence (AMI) is proposed for predicting time series data or data that are sensory in nature. The algorithm can learn to predict in a deterministic manner using a well defined formula; in fact the AMI does not necessarily require the tuning of any system specific parameter or hyper-parameter making it a very top candidate for real-time decision making and control functions.

AMI algorithm has been compared with two well known state-of-the-art algorithms for time series prediction: the Long Short-Term Memory (LSTM) and the Hierarchical Temporal Memory (HTM) based on cortical learning algorithms considering a number of related example cases. These algorithms have been reported by many authors to perform relatively well on a number of these time series examples; however, they could not surpass the AMI in all of the considered example cases. The reason for this is based on a hypothesized single reason: the AMI does not use the unnecessary weight or biasing operations as in most neural network processing schemes or in the case of cortical learning neural algorithms such as the HTM, the unnecessary column processing operations in the temporal parts to make a prediction. The AMI rather learns on the trend of the data using a novel formula based on sound neurobiological principles.

Despite its promising benefits, the potential of the AMI algorithm to make predictions multiple-steps ahead or based on multivariate based predictions in the context of anomaly detection remains fully unexplored; one possible approach to this may be to use the formulated state transition matrix in Phase-2 of the AMI to capture the trend in a sequence of *n-1* data points, then use the pattern formed from novel sequences for match comparisons that trigger a deviant state. We expect that academic researchers interested in novel prediction algorithms suitable for anomaly detection in sensory-like or time series data will explore this possibility or maybe an alternative but more efficient and less complex biological plausible approach.

The noise robustness of the AMI technique is also not studied in this research paper. Though the AMI features a sparsity parameter, *s*, the benefits are yet to be investigated; thus, this is another candidate area for research.

Another major shortcoming inherent in the AMI model used in this research revolves around the optimality of the threshold recognition parameter T_h for model adjustment. Thus, it will be advisable to investigate the potential of optimization algorithms that are bio-inspired such as the Bee Colony Optimization (BCO) or Particle Swarms (PS) for proper fitting of this parameter. Though, optimization will be beneficial, it also suggested that such optimizers should not be complex to implement in embedded hardware in order to facilitate real time predictive analytics.

References

Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, *262*, 134-147.

Cui, Y., Surpur, C., Ahmad, S., & Hawkins, J. (2016a). A comparative study of HTM and other neural network models for online sequence learning with streaming data. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 1530-1538). IEEE.

Cui, Y., Ahmad, S., & Hawkins, J. (2016b). Continuous online sequence learning with an unsupervised neural network model. *Neural computation*, *28*(11), 2474-2504.
Cui, Y., Ahmad, S., & Hawkins, J. (2017). The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding. *Frontiers in Computational Neuroscience*, *11*.

Cui, Y., Ahmad, S., & Hawkins, J. (2017). The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding. *Frontiers in computational neuroscience*, *11*, 111.

Goodfellow, I. (2016). *Deeplearning, MIT press*. doi:10.1016/B978-0-12-801775-3.00001-9.

Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to Forget: Continual Prediction with LSTM." *Neural Computation* 12, no. 10 (2000): 2451-2471.

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). IEEE.

Hawkins, J., Ahmad, S., & Dubinsky, D. (2010). Hierarchical temporal memory including HTM cortical learning algorithms. *Techical report, Numenta, Inc, Palto Alto.* *https://web.archive.org/web/20110714213347/http://www.numenta.com/htm-overview/education/HTM_CorticalLearningAlgorithms.pdf*

Hawkins, J., Ahmad, S., Purdy, S., & Lavin, A. (2016). Biological and machine intelligence (BAMI). *Initial online release 0.4.*

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*, *11*(10), 428-434.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Ivakhnenko, A. G. (1968). The group method of data of handling; a rival of the method of stochastic approximation. *Soviet Automatic Control*, *13*, 43-55.

Lavin, A., & Ahmad, S. (2015). Evaluating Real-Time Anomaly Detection Algorithms--The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 38-44). IEEE.

Liang, N. Y., Huang, G. B., Saratchandran, P., & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, *17*(6), 1411-1423.

Lieder, F., Daunizeau, J., Garrido, M. I., Friston, K. J., & Stephan, K. E. (2013a). Modelling trial-by-trial changes in the mismatch negativity. *PLoS computational biology*, *9*(2).

Lieder, F., Stephan, K. E., Daunizeau, J., Garrido, M. I., & Friston, K. J. (2013b). A neurocomputational model of the mismatch negativity. *PLoS computational biology*, *9*(11).

Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., & Stork, J. (2015). Comparison of different methods for univariate time series imputation in R. *arXiv preprint arXiv:1510.03924.*

Näätänen, R., Gaillard, A. W., & Mäntysalo, S. (1978). Early selective-attention effect on evoked potential reinterpreted. *Acta psychologica*, *42*(4), 313-329.

Näätänen, R., Paavilainen, P., Rinne, T., & Alho, K. (2007). The mismatch negativity (MMN) in basic research of central auditory processing: a review. *Clinical neurophysiology*,*118*(12), 2544-2590.

Osegi, E. N., & Anireh, V. I. (2016). Deviant Learning Algorithm: Learning Sparse Mismatch Representations through Time and Space. *arXiv preprint arXiv:1609.01459.*

Osegi, E. N., Anireh, V. I., & Onukwugha, C. G. (2018, June). pCWoT-MOBILE: a collaborative web based platform for real time control in the smart space. iSTEAMS SMART-MIINDs Conference, 13(3), 237-250.

Osegi, E.N., Taylor, O.E., Wokoma, B.A., & Idachaba, A.O. (2020, in-press). A Smart Grid Technique for Dynamic Load Prediction in Nigerian Power Distribution Network. International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering & Technology (ICSISCET- 2019), Gwalior, India

Rodriguez-Cobo, L., Ruiz-Lombera, R., Conde, O. M., López-Higuera, J. M., Cobo, A., & Mirapeix, J. (2013). Feasibility study of Hierarchical Temporal Memories applied to welding diagnostics. *Sensors and Actuators A: Physical*, *204*, 58-66.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011a). Semi- supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp.151-161). Association for Computational Linguistics.

Socher, R., Huang, E. H., Pennin, J., Manning, C. D., & Ng, A. Y. (2011b). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems* (pp. 801-809).

Sollini, J., Chapuis, G. A., Clopath, C., & Chadderton, P. (2018). ON-OFF receptive fields in auditory cortex diverge during development and contribute to directional sweep selectivity. *Nature communications*, *9*(1), 1-12.

Struye, J., & Latré, S. (2019, in-press). Hierarchical temporal memory and recurrent neural networks for time series prediction: an empirical validation and reduction to multilayer perceptrons. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2018.09.098

Sutskever, I., & Hinton, G. (2007). Learning multilevel distributed representations for high-dimensional sequences. In *Artificial intelligence and statistics* (pp. 548-555).

Takaura, K., & Fujii, N. (2016). Facilitative effect of repetitive presentation of one stimulus on cortical responses to other stimuli in macaque monkeys–a possible neural mechanism for mismatch negativity. *European Journal of Neuroscience*, *43*(4), 516-528.

Wokoma, B. A., & Osegi, E. N. (2019, October). A Resonant Fault Current Limiting Prediction Technique based on Auditory Machine Intelligence. In *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)* (pp. 1-5). IEEE.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

Yöntem, M. K., Adem, K., İlhan, T., & Kılıçarslan, S. (2019). Divorce prediction using correlation based feature selection and artificial neural networks. *Nevşehir Hacı Bektaş Veli Üniversitesi SBE Dergisi*, *9*(1), 259-273.

Appendix-1: Default parameters for the AMI, LSTM and HTM techniques

Table A.1: Default AMI Parameters

Parameter	Default value
Model Adjustment Threshold, T_h	0.21
Sparsity factor, *s*	2

Table A.2: Key LSTM Parameters

Parameter	Default value
Sequence Length	20
Hidden Neuron Size	100
Maximum Iteration	100
Maximum Epoch Size	100
Learning Rate	0.1

Table A.3: Key HTM Parameters

Parameter	Default value
Number of Columns	250
Initial Synaptic Permanence	0.21
Reduct factor	2
Boost factor	100
Synaptic Permanence Increment	0.1
Synaptic Permanence Decrement	0.1
Number of past sequences used as context	2