AJSE

American Journal of Science & Engineering

Volume 3 Issue 2

September 2022

American Journal of Science & Engineering (AJSE)
Society for Makers, Artists, Researchers and Technologists (SMART)
6408 Elizabeth Ave SE, Auburn 98092, Washington, USA
ISSN: 2687-9530 (Print) and 2687-9581 (Online)

Editor-in-Chief

Dr. Izzat Alsmadi

Texas A&M, San Antonio, USA

Research Interest: Cyber Intelligence, Cyber Security, Software Engineering, Social Networks

Bio: Izzat Alsmadi is an Assistant Professor in the department of computing and cyber security at the Texas A&M, San Antonio. He has his master and PhD in Software Engineering from North Dakota State University in 2006 and 2008. He has more than 100 conference and journal publications. His research interests include: Cyber intelligence, Cyber security, Software security, software engineering, software testing, social networks and software defined networking. He is lead author, editor in several books including: Springer The NICE Cyber Security Framework Cyber Security Intelligence and Analytics, 2019, Practical Information Security: A Competency-Based Education Course, 2018, Information Fusion for Cyber-Security Analytics (Studies in Computational Intelligence), 2016. The author is also a member of The National Initiative for Cybersecurity Education (NICE) group, which meets frequently to discuss enhancements on cyber security education at the national level.

Objective Assessment Application for Preschool Child Development

Guillaume Odendaal
Department of Mechanical and Mechatronic Engineering, Stellenbosch University, Stellenbosch, South Africa
Guillaume.odendaal@gmail.com

Anna Marie Clay
Agricultural and Biological Engineering, Mississippi State University, MS, USA
Amd406@msstate.edu

David Vandenheever
Agricultural and Biological Engineering, Mississippi State University, MS, USA
Department of Mechanical and Mechatronic Engineering, Stellenbosch University, Stellenbosch, South Africa
davidvdh@abe.msstate.edu

Abstract— Current developmental screening tests are typically subjectively evaluated making them susceptible to bias and are time- and resource-intensive. We present here the development of a tablet application for developmental screening incorporating fine motor and language tests. The tablet application was built with modularity in mind to ease the process of adaptation for cultural and age-appropriate conversions. An accompanying assessment pipeline was constructed to automatically process the data from the tablet assessment into several different metrics. The initial results indicate the usefulness and feasibility of the proposed application.

Keywords—early child development; language assessment; fine motor assessment; computerized assessment

I. Introduction

In order for a child to develop at the correct pace, he/she needs the appropriate stimulation and care; even from birth, they should be continuously stimulated at the appropriate level (Agyei, van der Weel, & van der Meer, 2016). The stimulation typically happens at home or in childcare facilities. For every child, the preschool years are the most important, because brain development and neural plasticity are at their peak (Chugani, 1998). Children that do not receive appropriate stimulation and care are at risk of not developing to their fullest potential. In 2016, the World Health Organization (WHO) estimated that 43% of children in low- and middle-income countries (LMICs) (250 million) were unable to reach their full development potential due to a lack of correct stimulation and care (WHO, 2018). The presence of neurodivergence in some children increases this risk. Neurodivergence is defined as a brain that functions in ways that diverge significantly from the dominant societal standards of "normal" and may include disorders such as Autistic Spectrum Disorder (ASD), Attention Deficit (Hyperactive) Disorder (ADD/ADHD), and dyslexia.

It has been shown that early intervention strategies can mitigate some of the effects of these disorders (Barnett, 1998), (Gorey, 2001). However, before interventions can be implemented, awareness of the problem needs to be obtained by the caregiver or guardians/parents. In LMICs, factors such as poverty, illiteracy of parents, and scarcity of resources can increase the lack of necessary stimulation and care children should receive (Engle & Black, 2008). These factors also contribute to the delayed discovery of neurodivergent disorders. There is therefore a need to assess several cognitive functions in young children to identify neural developmental issues and proceed with the necessary interventions.

There are several models of cognition defined by various bodies that each contain a set of cognitive domains (Harvey, 2019), (Baron & Leonberger, 2012), (Sabanathan, Wills, & Gladstone, 2015). Although there is no consensus on which model and accompanying domains are correct, five domains were present in all cognition models: attention, memory, executive functioning, language, and motor skills. Attention pertains to a person's ability to focus on and differentiate between important information and non-relevant information, and how long this focus can be held. Memory touches on all aspects that we think of as memory, which includes working memory, prospective memory, explicit memory, procedural memory, and semantic memory (Harvey, 2019). Executive functioning is a person's ability to execute a cognitive set (rules to follow when given a task, e.g., sorting cards according to color and not suit), mental flexibility, and inhibition. Language consists of two sub-domains, receptive and expressive language. The former is a person's ability to understand language such as instructions, and the latter is the ability to express or convey meaning/ideas. Motor skills also consist of two sub-domains, fine and gross motor. The former pertains to small movements of the fingers and hand, manipulation of objects, and typing/writing, whereas the latter refers to larger movements such as walking, sitting, standing up, balance, and physical strength.

The developed assessment application presented here focuses on two domains, fine motor and language. Although it is important to assess all cognitive domains to acquire a full picture, this is not always possible. Fine motor assessment, as opposed to overall motor skills, was chosen because the nature of gross motor assessments makes it difficult to assess using a tablet. Furthermore, motor skills are the first to develop within the cognitive domain (Casey, Tottenham, Liston, & Durston, 2005), and is therefore well aligned to test early development. Language was chosen as the second domain as it also develops early on before other higher cognitive functions such as executive functioning (Shonkoff, Boyce, Cameron, & et al, 2008).

Motor function and control mostly reside in the motor cortex of the brain. Motor function impairment has been found to accompany ADHD (Dewey, Cantell, & Crawford, 2007), (Pitcher, Piek, & Hay, 2003), dyslexia (Fawcett & Nicolson, 1995), and anxiety disorders (Erez, Gordon, Sever, Sadeh, & Mintz, 2004). Motor function impairment, also classified as a disorder known as Developmental Coordination Disorder (DCD), has been found to affect certain facets of life, such as mathematical skills, reading, and writing (Alloway, 2007). Longitudinal studies have also linked motor ability to executive functioning later in life (Murray, et al., 2006) and academic performance in mathematics (Kurdek & Sinclair, 2001). Motor function has also been shown to predict levels of anxiety and depressive symptomatologies (Piek, Barrett, Smith, Rigoli, & Gasson, 2010).

Language is an integral part of how people interact with the world and is a complex construct. The complex nature of language can be described as a system comprising of many dimensions, namely phonology (sound system), lexicon (vocabulary), semantics (meaning), grammar (structure), pragmatics (communicative functions and conventions for language use), and discourse (integration of utterances into longer stretches of conversation or narrative). The acquisition of language starts shortly after birth when infants start to discriminate between different sound contrasts (McMurray & Aslin, 2005). The prelinguistic period is characterized by speech sounds, babbling, and longer sequences of sounds trying to mimic adult speech (Saaristo-Helin, Kunnari, & Savinainen-Makkonen, 2011); this is followed by gestures, indicating wants and interactions (Behne, Liszkowski, Carpenter, & Tomasello, 2012), and finally basic language comprehension such as recognizing his/her name and associating words with objects (Tincoff & Jusczyk, 1999). Language acquisition speeds up once the first word is spoken, ending the pre-linguistic period. On average, children will acquire 10 words per month up to about 50 words, whereby this acquisition rate increases to about 30 words per month (Goldfield & Reznick, 1990). Two-word speech, indicating basic grammatical knowledge developing (Schipke & Kauschke, 2011), becomes more apparent and develops into three- and four-word utterances. Auxiliary verbs are next to develop, and only later in language development come questions and negative sentences (Tyack & Ingram, 1977). Finally, language development ends at the end of the child's preschool years (Hoff, 2009). Language deficiencies have been linked to various developmental and educational outcomes, such as the link with memory (Conti-Ramsden & Durkin, 2007), social behavior, quality of friendships (Durkin & Conti-Ramsden, 2007), emotional difficulties, and academic failure (St Clair, Pickles, Durkin, & Conti-Ramsden, 2011), (Conti-Ramsden, Durkin, Simkon, & Knox, 2009).

Developmental assessment is a wide field of study, having specific tests and procedures depending on the domain assessed. To simplify this, two categories of developmental assessment are considered: classical and computerized. Classical developmental assessment employs a test using pen and paper, usually administered by a trained medical professional. Alternatively, a computerized developmental assessment is an application with which the child interacts, which in turn calculates the necessary results. Computerized assessments are commonly given via tablets. Tablet technology is commonly preferred due to its lightweight and compact design (Kucirkove, 2014) and ability to quickly distribute newer versions of test development cycles. Furthermore, it has been shown that preschool children can successfully interact with tablet technologies (Nacher, Jaen, Navarro, Catala, & Gonzalez, 2015) as they are becoming increasingly popular in early education (Chiong & Shuler, 2010), (Geist, 2012).

Here we describe the development and validity of an in-house developed tablet assessment application to asses fine motor and language abilities.

II. Methods

A. Test items

A total of eighteen test items were implemented, ten language-related and eight fine motor-related. The process of selecting these tests started by collecting data from current standardized developmental assessments. Six computerized developmental assessments and fifteen classical developmental assessments (8 language-related, 4 fine motor-related, and 3 assessment batteries containing both language and fine motor assessments) were analyzed and filtered according to implementability, whether or not they would keep their construct validity when implemented on a tablet, and presence in the standardized tests. The list of developmental assessments that were used in the construction of this tablet application are listed for reference: Griffiths Mental Developmental Scales (GDMS), Clinical Evaluation of Language Fundamentals (CELF) (Wiig, Secord, & Semel, 2006), British Picture Vocabulary Scale - III (BPVS3) (Dunn & Dunn, 2009), Early Repetition Battery (ERB) (Seeff-Gabriel, Chiat, & Roy, 2008), Receptive One Word Picture Vocabulary Test (ROWPVT) (Martin & Brownell, 2010), Expressive One Word Picture Vocabulary Test (EOWPVT) (Martin & Brownell, 2010), Early Years Toolbox (EYT) (Howard & Melhuish, 2017), Denver Developmental Scales Test (Frankenburg, Dodds, Archer, Shapiro, & Bresnick, 1992), Zurich Neuromotor Assessment (ZNA), McCarron Assessment of Neuromuscular Development (MAND) (McCarron, 1976), Peabody Developmental Motor Scales 2 (PDMS-2) (Folio & Fewell, 2000), Peabody Picture Vocabulary Test - IV (PPVT-4) (Dunn & Dunn, PPVT-4 Peabody Picture Vocabulary Test, 2007), Bruininks-Oseretsky Test of Motor Proficiency 2nd edition (BOT-2) (Bruininks & Bruininks, 2005), and Movement Assessment Battery for Children (MABC-2) (Henderson, Sugden, & Barnett, 2007).

Using a Java-based Android Studio integrated development environment (IDE), images were created (using GIMP) and sourced (from Google, taking note of copyright and selecting only free-to-use images).

The first set of tests, known as option selection tests, focused on measurements of language skills. For this assessment, individuals are shown a stimulus, the stimulus needs to be understood, and the individual must respond by touching an option. Test items similar to these can be found in a variety of standardized developmental screening tests such as BAS3's, BPVS-3, ROWPVT-4, PPVT-4, EYT, DDST, and CELF. The following metrics are recorded for

each test: time to first response, time to correct response, and the number of options selected.

a) *Object Recall*
This is a short-term memory test in which an individual is shown a single object on the application's screen for a short period of time. Once the object is no longer visible, a grid of various objects, including the originally shown object, is presented. The individual is then asked to choose the original object.

b) *Choose Associated Word*
Here the individual is shown an object and four words. The individual is asked to select the word that best matches the object. If the individual desires to audibly hear a word, upon selection of the word, the tablet will audibly read it aloud.

c) *Choose Associated Object*
Here the individual is shown a word and four objects and is asked to select the appropriate object described by the word. Again, if the individual desires to audibly hear the given word, upon selection of the word, the tablet will read it aloud.

d) *Follow Instructions*
This item tests the individual's ability to follow instructions. The test begins with the tablet reading aloud a set of instructions to the individual. The individual is then allowed the opportunity to perform the given instructions. Examples of instruction sets for this test include, "select the cat on top of the table" (as opposed to the cat below the table), "select the cat on the left", and "select the cat on the right."

e) *Choose Picture*
This test item is an analysis of both receptive language skills and phonological memory. Individuals are given a description of an object and asked to select the correct object matching the description from four options. Examples include "green triangle" and "blue circle." This test item measures the individual's ability to recall in memory a few selectors (such as 'green' and 'triangle') but also understand the prompt given and connect it to a visual image.

The next set of test group assessments, labeled as placement accuracy, focused on measurements of fine motor skills. Present in many standard developmental screening tests such as DDST, BOT-2, PDMS, ZNA, MABC, and MAND, movements such as picking up, manipulating objects, and placing them in a desired position are generally used to measure fine motor ability. Although the actions of picking up and manipulating objects are not tested within this developmental screening application, the motor planning and on-screen manipulation aspects are still present. Furthermore, motor planning along with spatial intelligence is required to build structures or fit objects into place within a larger structure. The metrics recorded include the translation and rotation of all the puzzle pieces as well as the total time spent. The difference between the desired and completed coordinates and orientation of the puzzle pieces can be determined from this. We determined error in the X direction, error in the Y direction, Euclidian error, as well as rotation error. Distance errors were in pixels and rotation errors in degrees.

f) *Place Object Exactly*
In this test item, the individual is shown an incomplete object with the missing part shown to the side. The individual is tasked to move and rotate the missing part to fit and complete the shown object.

g) *Build Puzzle*
This test item is similar to building a puzzle. The individual is shown several pieces that fit together which must be rotated and moved to complete the image.

The next set of fine motor skill tests requires the individual to perform time-related tapping tasks. These are similar to standard developmental screening tests such as ZNA, MAND, and a tablet test [Pitchford and Outhwaite, 2016]. The test items involve tap-related tasks that are commonly used to measure manual processing speed and manual coordination. According to Avanzino et al., a person's ability to keep rhythm using motor movements is a predictor of fine motor ability (Avanzino, et al., 2016)

h) *Speed Tap*
Here the individual is tasked with tapping a dot displayed in the middle of the screen as quickly and accurately as possible in a set amount of time. This tests an individual's manual processing speed. The number of taps and the coordinates of the tap on the screen are recorded for accuracy.

i) *Rhythmic Tap*
This test assesses visuomotor coordination by evaluating how well the individual can match and continue a specific rhythmic beat. An auditory (1000 Hz beep for 300 ms) and visual (border of screen flashes between black and white) rhythmic beat is given for a predefined time before vanishing and the individual is asked to continue tapping the screen at the same rhythmic beat. The coordinates of the tap on the screen are recorded along with the timing which can be used to determine congruency with the given beat.

The next two motor skill assessment tests are based on two standardized line/path tracing-related tests known as MABC and BOT-2 as well as two standardized connecting dots-related tests, PDMS and BOT-2.

j) *Connect the Dots*
This is the classic connect the dots game where the individual is instructed to connect the numbered dots by tracing a straight line between two dots on the screen using their finger and starting at the dot numbered as one. For each segment (between two sequential dots) the perpendicular distance is determined between the drawn line and the correct line.

k) *Trace Path*
This test requires the individual to trace their finger along a predefined path. The correct path is divided into several points and the distance between each point and the nearest point on the drawn path is

determined. The time to complete the task is also recorded.

The next test is used to assess precision motor skills and is related to tests in PDMS and BOT-2.

l) Color Between the Lines

Here the individual is shown an image outline to color in, as well as a color palette to the side. The individual must color in between the lines, using a variety of stroke sizes. Measurements for post analysis includes the number of pixels that were required to be colored in, but were not, and the number of pixels that should not be colored, but were. Although not instructed to the individual, all pixels outside of the image's border and those that formed the image's outline were expected to remain unaltered. The overall score for the test is calculated as the percentage of error pixels.

Another image-based test requires individuals to copy and redraw a given picture. This test is based on similar tests in the DDST, PDMS and Griffiths.

m) Draw Object Shown

For this test the individual is presented with an image of an object on the left side of the screen and asked to redraw the object, using their finger, on the right side of the screen. Individuals are not constrained to draw the image in the exact orientation or scale. Since there is no accepted way of measuring how well the individual can redraw the image presented, six different metrics were evaluated: Sum of Squared Differences (SSD), Cosine Similarity (CS), Hausdorff Distance (HD) (Rucklidge, 1996), Scale Invariant Feature Transform (SIFT) (Lowe, 1999), a Convolutional Neural Network (CNN) based feature extraction method, and a machine learning based image similarity application protocol interface (API). The SIFT algorithm extracts keypoints from an image and descriptors of those keypoints. These keypoints and their accompanying descriptors are compared to the keypoints and descriptors of other images to find matches. The CNN feature extraction uses a modified pretrained ResCNN model, specifically ResNet-152. This model was specifically selected for its size and accuracy (He, Zhang, Ren, & Sun, 2016), (Anwar, 2019). The fully connected neural network at the end of the ResNet-152 network (that has the purpose of classifying images) was removed and the raw feature vectors were used to compare images. The final metric, DeepAI's Image, requires two images as the input and returns a similarity score as the output.

The last set of tests involve auditory processing. These tests involve the oral recall of numbers and sentences which is derived from related test items in BAS3, CELF, and Griffith; orally identifying the antonym of an image as based on similar tests in DDST and Griffiths; the oral description of a picture, which is an expressive language assessment, and based on tests in DDST, BAS3, and EOWPVT; and the inclusion of a novel computerized test item which assesses the pronunciation of a word. Medical professionals commonly look out for incorrect pronunciation when evaluating children undergoing an assessment battery. This is useful as language can be identified through this meta-analysis process (analysis of how the individual pronounces a word rather than what is heard). For all the tests in this set audio files were recorded and stored for post-processing. Two automatic speech recognition (ASR) techniques were used to make transcriptions of the audio recordings, namely DeepSpeech2 (Amodei, Ananthanaravanan, Anubhai, Bai, & et al, 2016), and Google's Speech-to-text API. Two metrics were used to assess the tests, word error rate (WER) and character error rate (CER). Word error rate allocated a point for each word in an individual's transcription that matched a word in the desired/correct transcription. Character error rate, a more lenient metric, calculated the number of changes needed in order for the two transcriptions to match (ie., the number of letters that need to be added, removed, or altered).

n) Number Recall

In this test a sequence of numbers is read out load. Following this, the individual is asked to repeat the sequence verbally. With each increasing trial, the sequence length increases by one number.

o) Sentence Recall

Similar to the previous test, in this test a sentence is read out load and the individual is asked to verbally repeat the test. Each subsequent trial will increase the sentence length by one word.

p) Give Opposite

In this test the individual is shown a word, the word is also read out loud, and the individual is then asked to verbally provide an antonym for the word.

q) Describe Picture

This test item shows a picture of a scene, and the individual is asked to verbally describe the scene. For this test item there is no original transcription with which to compare for correctness, and, therefore, three keywords were assigned to each picture. During post-processing analysis, each recording was checked for the number of keywords. If a keyword was present, a point was awarded for that picture. For example, if the individual was shown an image of a boy sitting on a chair reading a book. The three keywords associated with that image would be boy/person, sitting/sit, and read/reading.

r) Word Pronunciation

For the final test the individual is shown a word which is also read out loud. The individual is then asked to simply repeat the word.

B. Testing.

We tested each of the test items to see if we can clearly distinguish between good versus bad attempts. This serves as an initial validation test for each test item and does not constitute a methodologically sound experiment that will form part of future work. For the testing, one of the authors proceeded to perform the tests by intentionally simulating

good and bad attempts. In addition, we also performed these initial tests to determine which metric assessment techniques worked best for the *Draw Objects Shown* test item as well as for the auditory items.

III. Results

For test items a) through e) the application correctly detected whether the individual made the correct selection, how many selections were made, and the total time to selection. For test items f) and g), using different objects (and different numbers of puzzle pieces for g), we clearly showed measurable differences for all the error metrics for the simulated good versus bad attempts ($p<0.01$).

Figure 1 shows the results for test item h) with attempt 1 simulating the good attempt and attempt 2 simulating a bad attempt. The scenarios refer to the different times given to the individual to tap the screen starting with 10s and increasing by 5s to 30s. Figure 2 shows the results for test item i) again with attempt 1 simulating the good attempt and attempt 2 simulating a bad attempt. The scenarios refer to different rhythmic beats of 2 Hz, 1 Hz, 0.67 Hz, 0.5 Hz, and 0.4 Hz.

Figure 3 shows the error distance results for test item j) consisting of 12 segments. Similarly, figure 4 shows the mean error results for test item k) for three different line paths and for the good and bad attempts.

We were also able to distinguish between the two attempts for test item l) by calculating an error pixel percentage. Figure 5 shows an example with the good attempt giving a score of 97.7% and the bad attempt resulting in a score of 69.6%.

For test item m) the ResNet model was the only one that could reliably score the correctly drawn image higher than the two incorrectly drawn images.

For the various auditory test items, it was found that Google's Speech-to-text API worked best to transcribe the recordings.

IV. Discussion

Our initial results indicate the feasibility of the developed assessment application for preschool children. The application was developed to be modular (it is easy to add or remove tests) and easily modifiable (one can quickly and effortlessly change the specific pictures, words, or sentences used in the different test items). Our results indicate that it is possible to distinguish between good and bad attempts at the numerous different language and fine motor assessment items.

The first set of tests, collectively called the object selection tests, are used to assess language skills. A quick correct answer when making a selection gives an indication of the individual's receptive language skills. Faster processing speed with regard to language would allow the individual to perceive the stimulus and form an answer quicker (Leonard, et al., 2007). Furthermore, the selection of multiple answers can indicate hesitation or possible confusion which could warrant further investigation.

The next couple of tests (f and g), requiring the individual to move and rotate items, assess fine motor skills. Lower distance errors (where objects were placed closer to the desired locations) would correlate with better fine motor skills as the individual was better able to manipulate and translate the object(s) on screen. More specifically, test item g) gives insight into the individual's visuospatial intelligence as it is used to determine how pieces moved around and placed should fit together. More puzzle pieces require more spatial organization, therefore increasing the number of pieces indicates varying levels of visuospatial intelligence (Cameron, et al., 2012).

Both of the tapping tests (h and i) required the individual to tap the dot as accurately as possible. Here, the metric distance error indicates how far off the individual was from the dot's center. Lower scores would indicate better fine motor skills, such as visuomotor integration (using one's visual perception to guide where to tap on the screen). Furthermore, motor timing has a strong link to good overall motor performance (Falter & Noreika, 2011). A person's rhythmic capability indicates the ability to estimate time (sub- and supra-second) and uphold motor timing rhythm (the rhythm of movement and timing of movements). We also measured variance in the tapping frequency with higher variance indicating less consistency, which in turn could indicate motor timing ability (Noreika, Falter, & Rubia, 2013).

The two tracing accuracy test items (j and k) used distance error as a metric, with mean and variance calculated. Higher variance, average error, and overall distance error would indicate that the individual had traced the line (or between two dots) with less accuracy, thus indicating less fine motor precision and control. Therefore, less distance error would correlate with better fine motor ability (Cohen, Bravi, Bagni, & Minciacchi, 2018). Coloring an image (item l) requires fine motor control (Wehrmann, Chiu, Reid, & Sinclair, 2006), therefore better fine motor ability and control over fine movements would allow the individual to color the image more accurately and would result in fewer error pixels present. Furthermore, better fine motor control would allow the individual to redraw a picture more accurately (item m). Thus, the similarity score generated between the stock image (presented as a stimulus to the individual) and the drawn image can be used to indicate fine motor ability (Vimercati, et al., 2015). The higher the similarity score, the better the fine motor ability. The ResNet model was the only measure that consistently indicated that the intended drawn image was more similar than that of the intentionally incorrectly drawn images. This metric, however, still needs refinement.

Finally, the audio analysis test items are to be viewed with as much confidence as the confidence in the automatic speech recognition system. These systems can influence the results if they have not been specifically trained to avoid bias. The WER and CER metrics measureed how similar the spoken transcription is to the true transcription. Less WER and CER indicate a closer match between the two indicating that the individual understood the stimulus and objective. Therefore, lower WER and CER indicates better receptive language (Viding, et al., 2004).

There is a word of caution here. All stimuli (objects and words) used in this work were merely used as placeholders to demonstrate the capabilities of the application and accompanying processing pipeline. These stimuli might not suit all environments. The specific stimuli used can affect the results in several ways and are dependent

on the individual's familiarity with the stimulus, the individual's home language, whether the stimulus has an ambiguous meaning, or if the stimulus is too difficult for the individual's age. When choosing suitable stimuli for the test one should be guarded against biasing results through the use of specific stimuli and only cultural and age-appropriate stimuli should be used. Furthermore, the results derived by the processing pipeline are only valid if the individual understood what was expected of them. If the individual did not understand what is needed of them and performed the test item, the results would not yield an accurate representation of the construct being measured. The degree to which the individual understands the stimuli needs to be noted and taken into account when administering the test.

V. CONCLUSION

This work reviews the development and characterization of an automated tablet assessment application and its accompanying processing pipeline. This tablet application focused on two main domains, fine motor and language, specifically for preschool children. Eighteen test items were sourced, created, and implemented on a tablet application, and processing pipelines created for each of the test items. The results indicate that the application works as intended and is able to record multiple measures simultaneously. In publishing the characterization and results of the automated tablet assessment application design, it is hoped this application will serve as a framework for the development of objective assessment tests in the future.

REFERENCES

Agyei, S. B., van der Weel, F. R., & van der Meer, A. L. (2016). Longitudinal study of preterm and full-term infants: High-density EEG analysis of cortical activity in response to visual motion. *Neuropsychologia, 84*, 89-104.

Alloway, T. P. (2007). Working memory, reading, and mathematical skills in children with developmental coordination disorder. *Journal of Experimental Child Psychology, 96*, 20-36.

Amodei, D., Ananthanaravanan, A., Anubhai, R., Bai, J., & et al. (2016). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *Proceedings of the 33rd International Conference on Machine Learning.* New York.

Anwar, A. (2019). *Difference between AlexNet, VGGNet, ResNet, and Inception*. Retrieved from Towards Data Science: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96

Avanzino, L., Pelosin, E., Vicario, C. M., Lagravinese, G., Abbruzzesse, G., & Martino, D. (2016). Time Processing and Motor Control in Movement Disorders. *Frontiers in Human Neuroscience, 10*, 631.

Barnett, S. (1998). Long-term cognitive and academic effects of early childhood education on children in poverty. *Preventitive Medicine, 27*, 204-207.

Baron, I. S., & Leonberger, K. A. (2012). Assessment of intelligence in the preschool period. *Neuropsychology Reviews, 22*, 334-344.

Behne, T., Liszkowski, U., Carpenter, M., & Tomasello, M. (2012). Twelve-month-olds' comprehension and production of pointing. *British Journal of Developmental Psychology, 30*, 359-375.

Bruininks, R. H., & Bruininks, B. D. (2005). *BOT-2 Bruininks-Oseretsky Test of Motor Proficiency* (2nd ed.).

Cameron, C. E., Brock, L. L., Murrah, W. M., Bell, L. H., Worzalla, S. L., Grissmer, D., & Morrison, F. J. (2012). Fine Motor Skills and Executive Function Both Contribute to Kindergarten Achievement. *Child Development, 83*, 1229-1244.

Casey, B. J., Tottenham, N., Liston, C., & Durston, S. (2005). Imaging the developing brain: What have we learned about cognitive development? *Trends in COgnitive Sciences, 9*, 104-110.

Chiong, C., & Shuler, C. (2010). *Learning: Is there an app for that?* Joan Ganz Cooney Center.

Chugani, H. T. (1998). A critical period of brain development: Studies of cerebral glucose utilization with PET. *Preventitive Medicine, 27*, 184-188.

Cohen, E. J., Bravi, R., Bagni, M. A., & Minciacchi, D. (2018). Precision in drawing and tracing tasks: Different measures for different aspects of fine motor control. *Human Movement Science, 61*, 177-188.

Conti-Ramsden, G., & Durkin, K. (2007). Phonological short-term memory, language, and literacy: Developmental relationships in early adolescence in young people with SLI. *Journal of Child Psychology and Psychiatry, 48*, 147-156.

Conti-Ramsden, G., Durkin, K., Simkon, Z., & Knox, E. (2009). Specific language impairment and school outcomes. I: Identifying and explaining variability at the end of compulsory education. *International Journal of Language and COmmunication Disorders, 44*, 15-35.

Dewey, D., Cantell, M., & Crawford, S. G. (2007). Motor and gestural performance in children with autism spectrum disorders, developmental coordination disorder, and/or attention deficit hyperactivity disorder. *Journal of the International Neuropsychological Society, 13*, 246-256.

Dunn, L. M., & Dunn, D. M. (2007). *PPVT-4 Peabody Picture Vocabulary Test* (4th ed.).

Dunn, L. M., & Dunn, D. M. (2009). *The British picture vocabulary scale* (3rd ed.). GL Assessment Limited.

Durkin, K., & Conti-Ramsden, G. (2007). Language, social behavior, and the quality of friendships in adolescents with and without a history of specific language impairment. *Child Development, 78*, 1441-1457.

Engle, P. L., & Black, M. M. (2008). The effect of poverty on child development and educational outcomes. *Annals of the New York Academy of Sciences, 1136*, 143-356.

Erez, O., Gordon, C. R., Sever, J., Sadeh, A., & Mintz, M. (2004). Balance dysfunction in childhood anxiety: Findings and theoretical approach. *Journal of Anxiety Disorders, 18*, 341-356.

Falter, C. M., & Noreika, V. (2011). Interval Timing Deficits and Abnormal Cognitive Development. *Frontiers in Integrative Neuroscience, 5*, 1-2.

Fawcett, A. J., & Nicolson, R. I. (1995). Persistent deficits in motor skill of children with dyslexia. *Journal of Motor Behavior, 27*, 235-240.

Folio, M. R., & Fewell, R. R. (2000). *PDMS-2 Peabody Developmental Motor Scales* (2nd ed.).

Frankenburg, W. K., Dodds, J., Archer, P., Shapiro, H., & Bresnick, B. (1992). The Denver II: A major revision and

restandardization of the Denver Developmental Screening Test. *Pediatrics, 89*, 91-97.

Geist, E. (2012). A qualitative examination of two-year-olds interaction with tablet based interactive technology. *Journal of Instructional Psychology, 39*, 26-35.

Goldfield, B. A., & Reznick, J. S. (1990). Early Lexical Acquisition: Rate, Content, And The Vocabulary Spurt. *Journal of Child Language, 17*, 171-183.

Gorey, K. M. (2001). Early Childhood Education: A Meta-Analytic Affirmation of the Short- and Long-Term Benefits of Educational Opportunity. *School Psychology Quarterly, 16*, 9-30.

Harvey, P. D. (2019). Domains of cognition and their assessment. *Dialogues in Clinical Neuroscience, 21*, 227-237.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 770-778). Las Vegas, USA.

Henderson, S. E., Sugden, D. A., & Barnett, A. (2007). *Movement Assessment Battery for Children* (2nd ed.).

Hoff, E. (2009). *Language Development* (4th ed.). Belmont, CA: Wadsworth.

Howard, S. J., & Melhuish, E. (2017). An Early Years Toolbox for Assessing Early Executive Function, Language, Self-Regulation, and Social Development: Validity, Reliability, and Preliminary Norms. *Journal of Psychoeducational Assessment, 35*, 255-275.

Kucirkove, N. (2014).). iPads in early education: Separating assumptions and evidence. *Frontiers in Psychology, 5*, 715.

Kurdek, L. A., & Sinclair, R. J. (2001). Predicting reading and mathematics achievement in fourth-grade children from kindergarten readiness scores. *Journal of Educational Psychology, 93*, 451-455.

Leonard, L. B., Weismer, S. E., Miller, C. A., Francis, D. J., Tomblin, J. B., & Kail, R. V. (2007). Speed of processing, working memory, and language impairment in children. *Journal of Speech, Language, and Hearing Research, 50*, 408-428.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision*, (pp. 1150-1157). Kerkyra, Greece.

Martin, N. A., & Brownell, R. (2010). *EOWPVT-4: Expressive One-Word Picture Vocabulary Test* (4th ed.).

Martin, N. A., & Brownell, R. (2010). *ROWPVT-4: Receptive One-Word Picture Vocabulary Test* (4th ed.).

McCarron, L. T. (1976). *MAND : McCarron assessment of neuromuscular development, fine and gross motor abilities.* Common Market Press.

McMurray, B., & Aslin, R. N. (2005). Infants are sensitive to within-category variation in speech perception. *Cognition, 95*, B15-26.

Murray, G. K., Veijola, J., Moilanen, K., Miettunen, J., Glahn, D. C., Cannon, T. D., . . . Isohanni, M. (2006). Infant motor development is associated with adult cognitive categorisation in a longitudinal birth cohort study. *Journal of Child Psychology and Psychiatry and Allied Disciplines, 47*, 25-29.

Nacher, V., Jaen, J., Navarro, E., Catala, A., & Gonzalez, P. (2015). Multi-touch gestures for prekindergarten children. *International Journal of Human Computer Studies, 73*, 37-51.

Noreika, V., Falter, C. M., & Rubia, K. (2013).). Timing deficits in attention-deficit/hyperactivity disorder (ADHD): Evidence from neurocognitive and neuroimaging studies. *Neuropsychologia, 51*, 235-266.

Piek, J. P., Barrett, N. C., Smith, L. M., Rigoli, D., & Gasson, N. (2010). Do motor skills in infancy and early childhood predict anxious and depressive symptomatology at school age? *Human Movement Science, 29*, 777-786.

Pitcher, T. M., Piek, J. P., & Hay, D. A. (2003). Fine and gross motor ability in males with ADHD. *Developmental Medicine and Child Neurology, 45*, 525-535.

Rucklidge, W. (1996). *Efficient visual recognition using the Hausdorff distance.* Berlin: Springer.

Saaristo-Helin, K., Kunnari, S., & Savinainen-Makkonen, T. (2011). Phonological development in children learning Finnish: A review. *First Language, 31*, 342-363.

Sabanathan, S., Wills, B., & Gladstone, M. (2015). Child development assessment tools in low-income and middle-income countries: How can we use them more appropriately? *Archives of Disease in Childhood, 100*, 482-488.

Schipke, C. S., & Kauschke, C. (2011). Early word formation in German language acquisition: A study on word formation growth during the second and third years. *First Language, 31*, 67-82.

Seeff-Gabriel, B., Chiat, S., & Roy, P. (2008). *Early Repetition Battery.* London: Pearson Assessment.

Shonkoff, J. P., Boyce, T. W., Cameron, J., & et al. (2008). *The Timing and Quality of Early Experiences Combine to Shape Brain Architecture.* Retrieved from The National Scientific Council on the Developing Child: http://www.developingchild.net

St Clair, M. C., Pickles, A., Durkin, K., & Conti-Ramsden, G. (2011). A longitudinal study of behavioral, emotional and social difficulties in individuals with a history of specific language impairment (SLI). *Journal of Communication Disorders, 44*, 186-199.

Tincoff, R., & Jusczyk, P. W. (1999). Some beginnings of word comprehension in 6-month-olds. *Psychological Science, 10*, 172-175.

Tyack, D., & Ingram, D. (1977). Children's production and comprehension of questions. *Journal of Child Language, 4*, 211-224.

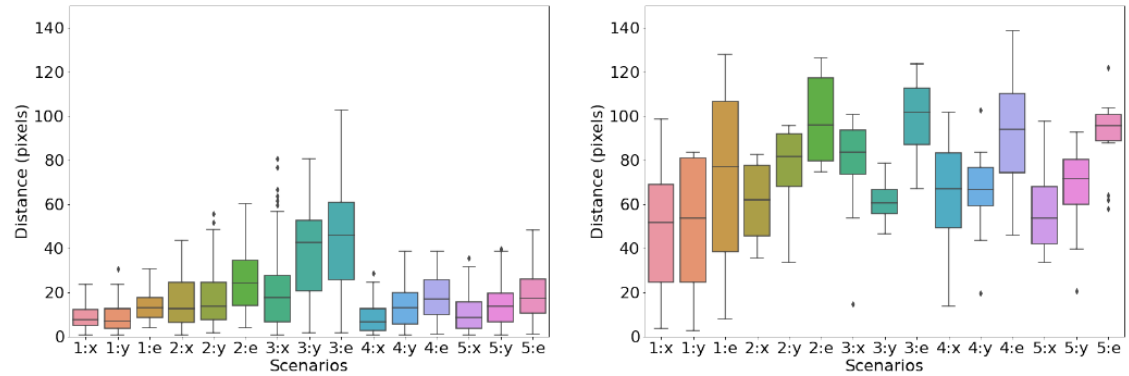
Viding, E., Spinath, F. M., Price, T. S., Bishop, D. V., Dale, P. S., & Plomin, R. (2004). Genetic and environmental influence on language impairment in 4-year-old same-sex and opposite-sex twins. *Journal of Child Psychology and Psychiatry, 45*, 315-325.

Vimercati, S. L., Galli, M., Stella, G., Caiazzo, G., Ancillao, A., & Albertini, G. (2015). Clumsiness in fine motor tasks: Evidence from the quantitative drawing evaluation of children with Down Syndrome. *Journal of Intellectual Disability Research, 59*, 248-256.

Wehrmann, S., Chiu, T., Reid, D., & Sinclair, G. (2006).). Evaluation of occupational therapy schoolbased consultation service for students with fine motor difficulties. *Canadian Journal of Occupational Therapy, 73*, 225-235.

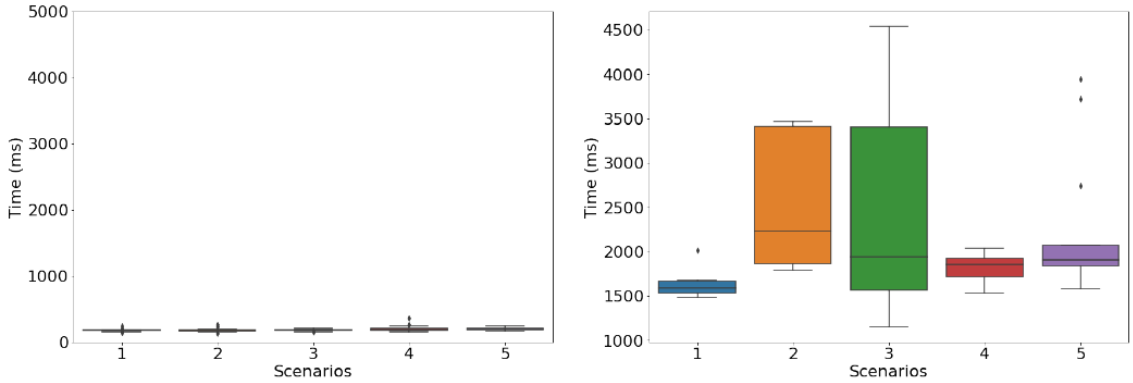
WHO. (2018). *Improving early childhood development: WHO guideline.* WHO.

Wiig, E. H., Secord, W., & Semel, E. M. (2006). *CELF Preschool 2 UK: Clinical Evaluation of Language Fundamentals, Preschool*. Harcourt Assessment.



(a) Attempt 1 distance error variance.

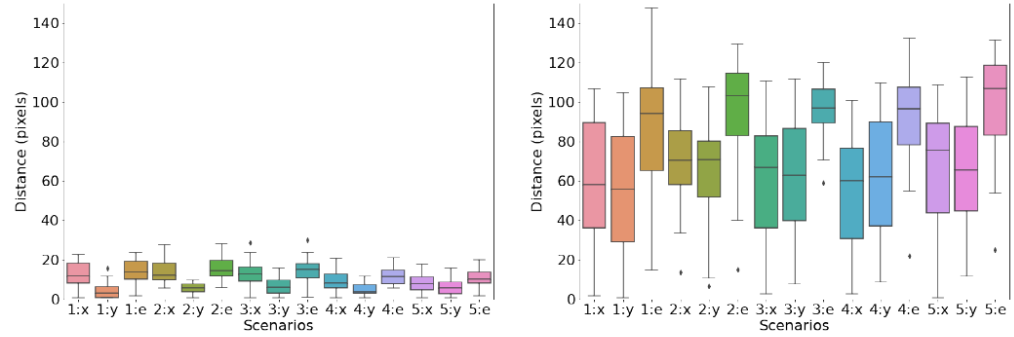
(b) Attempt 2 distance error variance.



(c) Attempt 1 inter tap time variance.

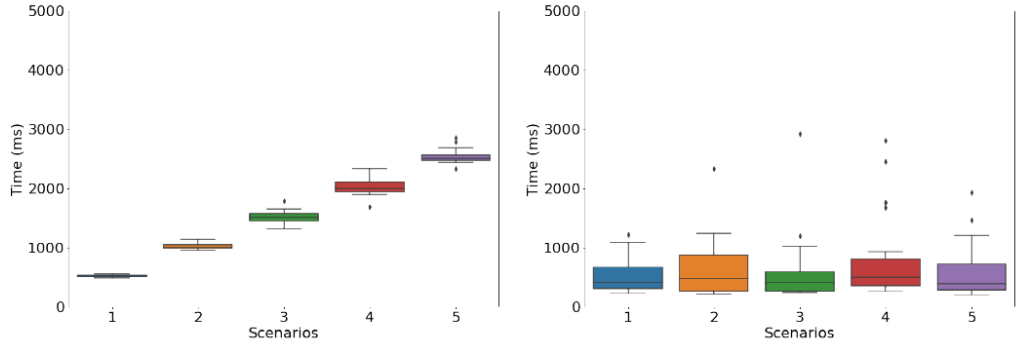
(d) Attempt 2 inter tap time variance.

Fig. 1. Results of test item h) where a) and b) shows the mean x, y, and Euclidian distance errors from the middle of the target for attempt 1 and 2, respectively, and c) and d) show the inter-tap variability for attempt 1 and 2, respectively.



(a) Attempt 1 distance error variance.

(b) Attempt 2 distance error variance.



(c) Attempt 1 inter tap time variance.

(d) Attempt 2 inter tap time variance.

Fig. 2. Results of test item i) where a) and b) shows the mean x, y, and Euclidian distance errors from the middle of the target for attempt 1 and 2, respectively, and c) and d) show the inter-tap variability for attempt 1 and 2, respectively.

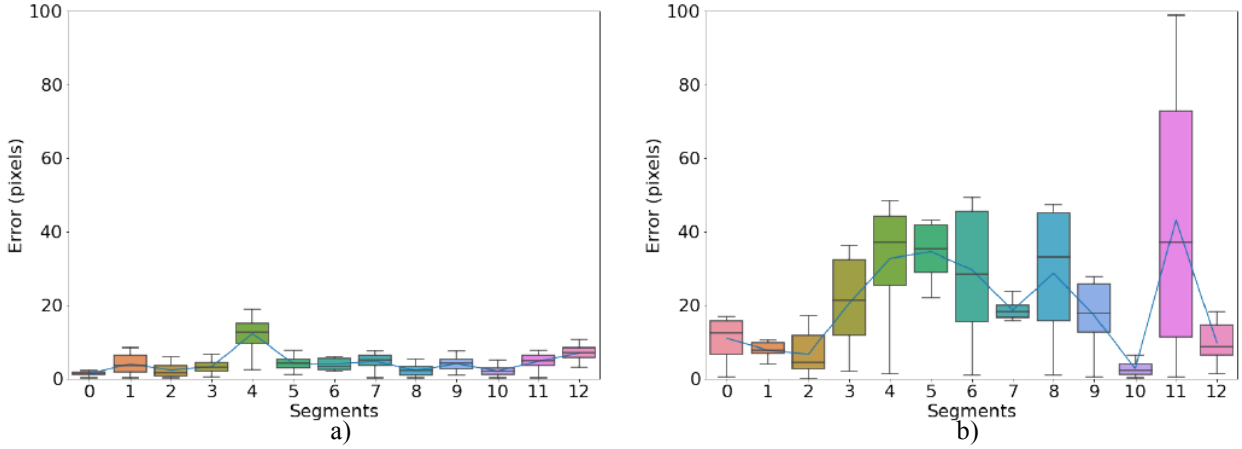


Fig. 3. Perpendicular distance (in pixels) of the line segment drawn from the optimal line for a) good attempt and b) bad attempt.

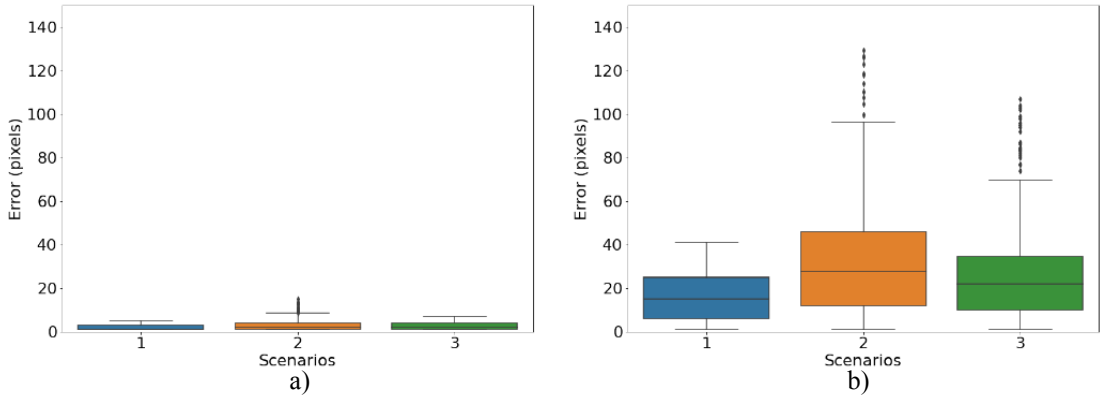
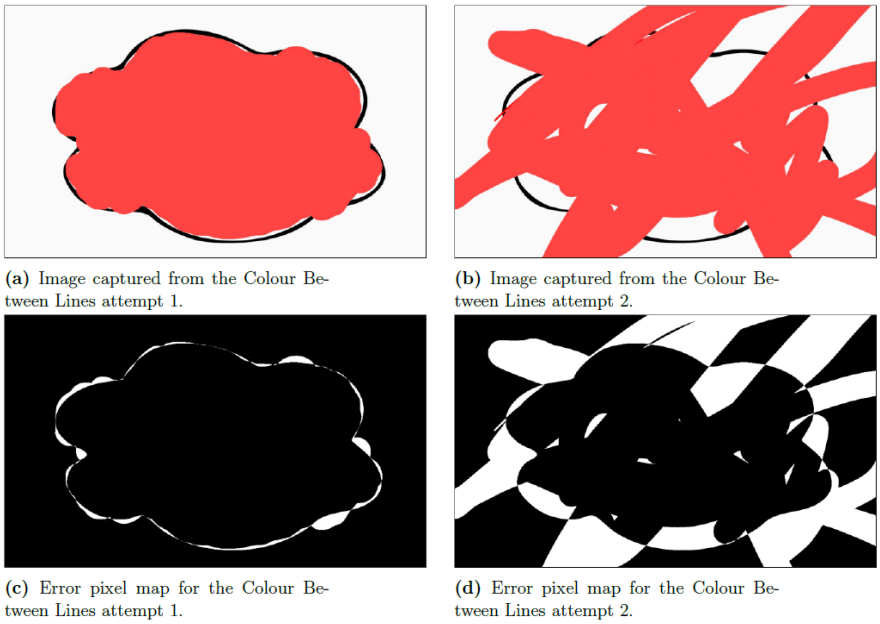


Fig. 4. Mean error distance for test item k) for three scenarios, a) simulates a good attempt and b) simulates a bad attempt.



(a) Image captured from the Colour Between Lines attempt 1.

(b) Image captured from the Colour Between Lines attempt 2.

(c) Error pixel map for the Colour Between Lines attempt 1.

(d) Error pixel map for the Colour Between Lines attempt 2.

Fig. 5. Results for test item l) where a) attempt 1 simulates the good attempt and b) attempt 2 simulates the bad attempt. The error map is shown in c) for attempt 1 and d) for attempt 2.

BEHAVIOR ALGORITHM — A NOVEL TIME-SERIES CLUSTERING APPROACH

Shaho Alaee
George Washington University
ORCID: 0000-0002-4735-6552

Jason M. Pittman
University of Maryland Global Campus
ORCID: 0000-0002-5198-8157

Abstract- Clustering time-series values is an established technique for organizations employing machine learning to analyze temporal datasets. Generally speaking, the goal of time-series methodology is to generate predictions. Such predictions could help organizations understand potential future cyberattacks, financial market changes, weather, or disease outbreaks. However, computational limitations lead existing algorithms to fail to group individual series together based the actual behavior of the series. A feature that can be used or derived to explain the time-series behavior had not been identified in the literature despite there being a need to have numeric values to describe the pattern of values over time. To address this gap, this work presents a behavior algorithm which addresses clustering time-series data based solely on the behavior of the series. Further, the algorithm is designed to operate effectively regardless of absolute values or temporal shifts. First, we describe the algorithm through mathematical examples. We provide the design approach for the algorithm numerically and through data visualizations. Then, we validated the algorithm on sample random data. Finally, we offer conclusions along with notions for future work based on this study.

Keywords- time-series · clustering · behavior · algorithms

1. INTRODUCTION

A variety of industries such as cybersecurity [1], human performance [2], finance [3], and medicine [4] leverage time series analysis. The essential purpose of employing any form of time-series methodology is to mine a prediction out of a diverse dataset. These predictions have great value to society: cyberattacks, financial market predictions, weather forecasting, and disease outbreak management to mention a few timely examples. Given the social impact of time-series analysis, the value to science and technology is in the creation, implementation, and optimization of these algorithms[5].

A fundamental approach to analyzing such data is clustering [6], especially in the context of temporal data. However,existing algorithms for analyzing time-series data fail to quantify and visualize how different series compare basedon temporal fluctuations [7, 8, 9]. Granted, fluctuations occur in most data samples and are unavoidable [10, 11]. Yet,this leaves a gap in which clustering categorical data by behavioral patterns is computationally expensive and does not provide insight beyond how a single series of data is related to another from a time-series perspective (e.g., a series that has high fluctuation or variation versus low).

This problem is significant to future forecasting efforts because such patterns aid in deeper understanding of the data, unforeseen similarities and differences across the dataset, clustering data points in a new dimension, and computational time savings when comparing time series data. Accordingly, this work addresses the lack of behaviorally adjusted time-series models by first identifying numeric features representing the behavior of the series. We then describe a novel algorithm aimed at improving the way time-series clustering and classification is conducted.

It is necessary to first establish the foundation related to this work. The next section offers a robust background of literature associated with time-series analysis. Particular areas of focus include research problems, the solutions offered throughout the history of the field, as well as synthesis of features across the research base important to the novel algorithm proposed in this study.

2. RELATED WORK

Time-series clustering is not new. Organizations base forecasting on time-series data through trend and seasonality to compute values such as rolling forecasts [12]. Standard forecasting techniques include analysis of trend and seasonality [13]. This approach is used to analyze a series and make predictions but does not provide a way to compare two series. For example, looking at a customer's expenditure and predicting their expenditures for the following year, or by grouping all customers together for a given country and predicting the following year. This led to the creation of other approaches with the goal of clustering

time-series data. While many approaches currently exist in modeling and working with time-series data, a clustering method that looks at behavior did not [7, 8, 9, 14, 15, 16].

This section explores the conceptual framework of time-series analyzes. The conceptual framework emphasizes the strengths of four approaches representing the state of the art in terms of time-series clustering. To that end, Aghabozorgi, Seyed, and Wah [17] gathered time-series approaches that were used in the last decade. The authors identified the most common approaches as The Hausdorff distance, HMM-based distance, Dynamic Time Warping (DTW), Euclidean distance, short times series distance, and Longest Common Sub-Sequence (LCS).

In general, clustering techniques measure similarity of some sort and those which are closer in some space (i.e., a plane or vector) are more similar [18]. Each of the following algorithms also aim to measure similarity through distance, however with different approaches. Using distance as an approach does have success but exhibit exponential time complexity and thus increasing computational expense. This is due to the fundamental approach in how distance is measured.

First, all series have x amount of data points, where x is the number of values in the data set for a single series. Then, every series is measured against every other series. The output is a specific series or ID and a list of all other series or IDs that are similar based on proximity by measuring the distance. For instance, if a data set has ten different time-series, each with ten different time points with a corresponding value, all time-series clustering algorithms would take every series and measure the distance for use as a central vector for clustering. A trivial amount of data is not a problem; however, in real world applications the data set will become too large, rendering this approach infeasible. Additionally, this does not include the issue of the finding similar behavior between series', regardless of the values.

2.1 Euclidean Distance

Euclidean distance has high accuracy as it measures the distance between all the values [19]. Two limitations of this algorithm is that values need to be the same length (i.e., same dimensions) and the series need to be aligned temporally [17]. For example, imagine two customers spending money every month for one year. Measuring the Euclidean distance between the two would compare January to January, February to February, March to March, so on and so forth. This is okay if the goal is to compare expenditure with a fixed time variable. However, if the patterns were similar regardless of when expenditures happened, this approach wouldn't work. Put simply, calculating the Euclidean distance excels in finding series that have the same pattern at the same time points, when peaks and valleys happen at the same timestamps [14]. In contrast, when the series have non-matching fluctuations and do not match, it is very likely that their scores will indicate no similarity.

2.2 Dynamic Time Warping

DTW is an approach to taking time-series and finding similarities as opposed to identical time alignments as seen with Euclidean distance [20]. DTW was first created to identify similar sounds waves where high accuracy was important in trade-off of high time complexity [21]. Similar, non-music based problems can be handled by DTW as the algorithm deals well with time-series clustering in the face of temporal shifts [17] and works well on small data sets. However, the algorithm is limited by its space and time complexities [22, 23]. The reason behind the extensive computation is that every timestamp in the series has the difference measured with every timestamp of every other series. Different DTW approaches have been researched to overcome the challenge of computational limitations on large data sets [20]. Finally, DTW requires all series in the data set to be of equal sample size [17]. Consequently, when an exact match is not as vital based on the data set the computation limitations may not be worth the output or feasible depending on the application. Improvements have been made to reduce the time complexity, such as Mini-DTW, which aims to summarize a dataset so that it isn't as large [23]. However, this doesn't solve the gap of find similar patterns across different time-series without comparison.

2.3 Short Time-Series

The short time-series distance approach to clustering accounts for shorter temporal lengths in the data [17, 18]. This time-series algorithm clusters exact behaviors based on the time-series shape [24]. Such an approach is limited to clustering short time-series and exhibiting the same patterns [14]. Furthermore, short time-series cannot group different patterns together exhibiting the same behavior or fluctuations [17, 18]. At the same time, short time-series is resilient to absolute values and will not skew the data behavior if such are included in the input.

2.4 Level Shift Detection

Level Shift Detection (LSD) is an established time-series algorithm applicable to datasets containing a nontrivial quantity of outliers and anomalies [25]. LSD is applied in one of two forms: specific frequencies of patterns greater than a threshold or to find when the average over time changes in different segments of the series [26]. Similar series can be clustered by counting

the number of occurrences for specific patterns [17, 14]. When looking at the second application where averages in certain windows change, LSD aims to measure how a series adjusts over time [27], not necessarily to cluster similar series together.

3. THE BEHAVIOR ALGORITHM

Current approaches to clustering focus on absolute values, which group series together where values are more similar [17]. These clustering mechanisms ignore the behavior of the series. Thus, the first step in using the behavior algorithm is to compute the sum of absolute difference and sum of series approach, after the data is normalized. It should be noted that using absolute values could cause large differences in distance measures depending on the variation of values that lie in the data set. Normalizing the data could find similarities of behavior by taking the effects of absolute values out of the equation regardless of the algorithm used. This was also demonstrated by [28] where taking the Euclidean distance of normalized data outperforms other algorithms and itself on absolute values.

The following (Figure 1) is an example of how absolute values and some standard descriptive statistics fail to identify patterns in time-series data. Series A has a value of 48,000 once, while series B has a value of 4,000 twelve times. The series averages and totals would be equal except the behavior was different between series A and B. The failure to identify patterns is caused by calculating the average and total amount and using them as metrics for distinction. It is not possible to conclude a difference in the behavior with such statistics.

Figure 1: The discrepancy based on fluctuation behavior in time-series data

Standard deviation and the variance outline the fluctuation behavior in the time-series behavior (Table 1). However, if standard deviation and variance can identify changes in the series behavior, then a new algorithm is not needed to group similar customers. Importantly, standard deviation and variance do not account for significantly different absolute values.

Table 1: Time-Series Descriptive Statistics

	Series A	**Series B**
Example	48,000 x 1	4,000 x 12
Average	4,000	4,000
Total	48,000	48,000
Std Dev	13,856	0
Variation	192,000,000	0

To extend the previous example demonstrating why absolute values will inaccurately handle fluctuation behaviors, we can visualize Series A with a value of 48,000 once compared to Series C with a value of 15,000,000 once (Figure 2). This example demonstrates two series where the behavior is the same but the standard deviation and variance are different. The visualization also shows behavior that is not easy to distinguish when plotted on the same graph with absolute values.

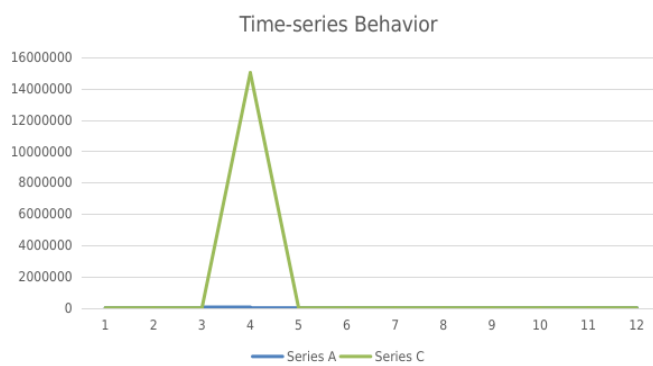


Figure 2: Visualizing hard to distinguish time-series behavior

Table 2: Descriptive Statistics for Two-Series Behavior

	Series A	**Series C**
Example	48,000 once	15,000,000 once
Average	4,000	1,250,000
Total	48,000	15,000,000
Std Dev	13,856	4,145,781
Variation	192,000,000	17,187,500,000,000

Based on the absolute values, series A and B had more in common because the average and total were the same. However, the actual behaviors of series A and C were more alike. Accordingly, the behavior algorithm must handle this discrepancy by normalizing the data first either by computing the traditional normalization technique or standard score (z-score). Further, absolute values make it difficult to group data elements based on their behavior which must beaccounted algorithmically.

For clarity, we take normalization to be a process of converting all values in a data list to values between zero and one. The normalization equation can be expressed as:

$$N_i = (X_i - X_{min})/(X_{max} - X_{min}) \quad (1)$$

The z-score is a way to determine how many standard deviations (σ) a specific data point is from the mean of the dataset. The standardization (z-score) equation can be expressed as:

$$Z_i = (X_i -,)/\sigma \quad (2)$$

Table 3: Descriptive Statistics for Temporal Shifts

	Series A			Series B		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	21.00	1.00	0.00	111.00	1.00	0.00
STD	2.49	0.28	1.00	27.36	0.28	1.00
VAR	6.19	0.08	1.00	748.69	0.08	1.00
Mean	1.75	0.08	0.00	9.25	0.08	0.00
Median	1.00	0.00	-0.30	1.00	0.00	-0.30
Sum of Abs Diff	9.00	1.00	3.62	99.00	1.00	3.62

To account for the issue where the standard deviation is zero ($\sigma = 0$), the z-score equation can be modified to include epsilon (ϵ). Epsilon is a very small constant, much smaller than any meaningful standard deviation, to avoid a *division by zero* error. The modified standardization (z-score) with epsilon equation can be expressed as:

$$Z_i = (X_i -,)/(\sigma + \epsilon) \quad (3)$$

Using either approach places every data element on the same scale. Notably, the ϵ modified approach handled a large difference in the total amount between series leading to concomitant difference in the standard deviation and variances. Such large differences render clustering difficult, if not impossible. This becomes more evident when all three-series are plotted with standard normalized values (Figure 3).

Figure 3: Visualizing clustering difficulties with three-series normalized values.

Series A and C are shifted to different time points for visibility but otherwise are identical. This data illustrates the twochallenges of (a) identifying a way to handle the behavior of series regardless of absolute value; (b) and the difference of time (i.e., temporal shifts) serving as a confounding factor. Visually, we can see that normalization accounts for absolute values and acts a key step in the behavior algorithm's computation of the sum of absolute difference and sum of the series.

Series A and B (Table 3) both have values of one for eleven out of twelve timestamps. Series A has one value of ten whereas series B has one value of one hundred. The behavior is the identical. However, looking at the standard statistics, it is difficult to find the similarity. The sum of absolute difference and sum of series of the z-scores and normalizationare the same for both series. Variance and standard deviation on the normalized data also appear to correctly identify the similarity in the pattern.

Table 4: Descriptive Statistics for Uneven Series Data

	Series C			Series D		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	120.00	12.00	0.00	1200.00	12.00	0.00
STD	0.00	0.00	0.00	0.00	0.00	0.00
VAR	0.00	0.00	0.00	0.00	0.00	0.00
Mean	10.00	1.00	0.00	100.00	1.00	0.00
Median	10.00	1.00	0.00	100.00	1.00	0.00
Sum of Abs Diff	0.00	11.00	0.00	0.00	11.00	0.00

Series C and D (Table 4) both have the same values for every point out of twelve in their respective series. Series C hastwelve values of ten whereas series D has twelve values of one hundred. The behavior is the same, however looking at the standard statistics, the similarity is obscured. The sum of absolute difference and sum of series of the z-scores and normalization are the same for both series. Variance and standard deviation on the normalized data also appear to correctly identify the similarity in the pattern.

Table 5: Descriptive Statistics for Dissimlar Time-Series

	Series E			Series F		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	48.00	6.00	0.00	240.00	6.00	0.00
STD	1.63	0.41	1.00	8.16	0.41	1.00
VAR	2.67	0.17	1.00	66.67	0.17	1.00
Mean	4.00	0.50	0.00	20.00	0.50	0.00
Median	4.00	0.50	0.00	20.00	0.50	0.00
Sum of Abs Diff	30.00	7.50	18.37	150.00	7.50	18.37

Series E and F (Table 5) fluctuate between three different values. Both have each value appear four different times in a specific order following the pattern *123123123123*. The behavior is the same, however looking at the standard statistics, it is again difficult to find the similarity. The sum of absolute difference and sum of series of the z-scores and normalization are the same for both series. Variance and standard deviation on the normalized data also appear tocorrectly identify the similarity in the pattern.

To demonstrate that sum of absolute difference and sum of series do not just consider the values at the time points, whether absolute or normalized, another example is provided (Table 6). Series G and H are two series which have thesame values but were recorded in a different pattern.

Table 6: Descriptive Statistics for Pattern Variance in Time-Series

	Series G			Series H		
	Actual	Normalized	Z-Score	Actual	Normalized	Z-Score
Sum of Series	90.00	6.00	0.00	90.00	6.00	0.00
STD	2.50	0.50	1.00	2.50	0.50	1.00
VAR	6.25	0.25	1.00	6.25	0.25	1.00
Mean	7.50	0.50	0.00	7.50	0.50	0.00
Median	7.50	0.50	0.00	7.50	0.50	0.00
Sum of Abs Diff	55.00	11.00	22.00	5.00	1.00	2.00

Series G and H have the same values yet the pattern is different. Series G alternates between two values for every timestamp, whereas H has the same value for the first six timestamps. Then, Series H changes to a different value for the following six measures. Thus, the behavior is different. Standard statistics, such as standard deviation, variance, mean, and median, are identical. The sum of absolute difference of the z-scores and standard normalization express numerically the behavior is in fact different. However, the sum of series is the same for both as it does not account for the order of which the expenditure happened. The standard deviation and variance also identified similarity in patterns.Notably, when applied to a larger sample size, limitations start to arise.

To that end, Figure 4 visualizes the different data for series A (i.e., original, normalized, and z-score). A key takeaway from the visualization is the behavior of the series is still intact. These results point towards normalization being necessary either through traditional normalization or computing the z-score as the new scores do not alter the behaviorof the series.

4. Discussion

The z-score provides a type of normalization as well. A strength of the z-score is the sum of absolute difference adequately finds similarities in series where behaviors were identical. On the other hand, a limitation is summing the computed z-score (i.e., computing the sum of series), produces a total which is always zero due to the negative numbers. Therefore, the sum of series does not provide any level of distinction in those values. The standard deviation and variance calculated on the z-scores computed to one for every element, with a mean of zero. With variance and

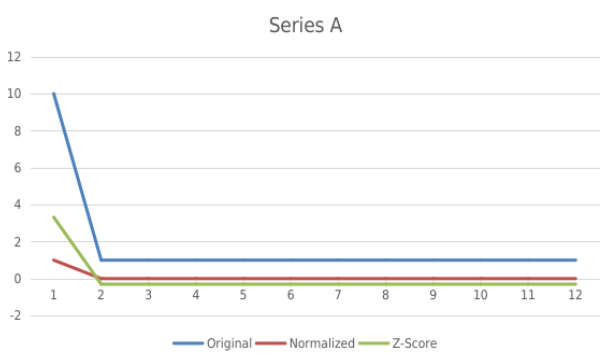


Figure 4: Visualizaing normalization of time-series data for time-series integrity.

standard deviation identified as attributes changing with high fluctuation in behavior, the z-score variance and standarddeviation lose their meaning.

The standard normalization's sum of absolute difference and sum of series also adequately compute scores that are reflective of their behavior and not absolute values. The additional benefit is that the sum of series also provides a level of distinction between different behaviors and a score that is the same for identical behaviors. The mean and median provide some insight as they also identify similarity in the behavior. These values could also match when behavior is significantly different due to how they are computed. Standard deviation and variance also provide a level of distinction between the different series in the fictional data set.

This highlights the importance of normalizing the data to find similarities in behavior. Normalizing the z-scores does turn the data into the exact values that normalizing the total amounts gets without the extra calculation of computing the average, standard deviation, to then compute the z-score so that it can be normalized to get the same results. It is recommended to normalize the actual values as the principle of parsimony dictates that the simpler model should be chosen when there is little benefit to the more complex one. In this case the outputs derived are exactly equal, thereforeforgoing the extra steps is best practice.

Furthermore, not only does the sum of absolute difference help prepare the series data to be clustered based on similarity, the values are interpretable. A value of zero, or close to zero, will occur when all values are the same or if the series has a small sample. A value of one suggests all the values were the same except for one. This can occur for any length of time-series. For example, a series with two data points will equal one as well as a series with three data points where two are the same value. When values are greater than one, the larger the sum of absolute difference, the closer the data points are together in the series. Series E and F demonstrate this point. The smaller the sum of absolute difference, the larger the gap the between the highest value and the rest. Series A and B demonstrate this point. Depending on the pattern, the sum of absolute difference can identify a specific behavior that the sum of series cannot. The sum of absolute difference unique example demonstrates this point (Figure 5).

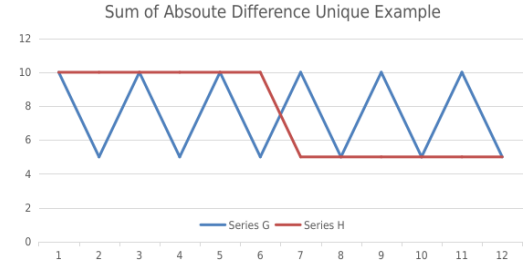


Figure 5: Visualizing patterns of specific behaviors based on absolute difference.

Here, Series G and H would have a sum of series value of six where no distinction could be made, whereas the sum of absolute difference would be one for series H and eleven for series G. Understanding this upfront can help determinewhich value is important to use and what the interpretation implies. The sum of series provides similar insight, howeverone aspect differs from sum of absolute difference. The sum of series exhibits the same behavior at both the minimum and maximum values (i.e., a drastic incline or decline). The values towards the middle exhibit a more regular behavior. The lower the sum, the more variation the series has while the higher the sum, the lower the variation until it reverses back to more variation (Figure 6).

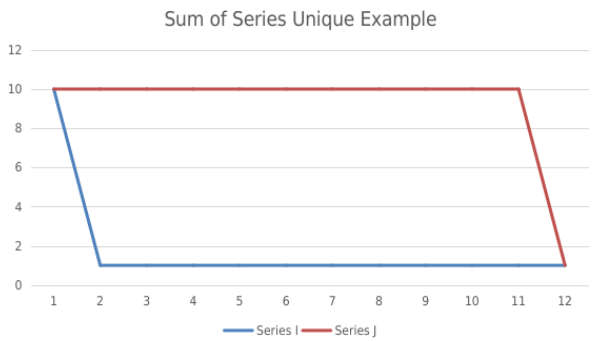


Figure 6: Visualizing variation in time-series with fluctuation behaviors.

The sum of absolute difference in both series would be the same- one- whereas the sum of series would be higher forseries J and lower for series I. Series I would have a sum of series value of 1 and series J would have a sum of series value of eleven.

4.1 Interval Selection

Feature engineering is a key step to optimize time-series output for interpretability. To that end, a data preprocessing step can help regardless of the absolute values. The feature is interval selection. Depending on the appropriate choice,data then needs to be aggregated for that interval. For example, if daily data is available, but the desired interval is monthly, then the daily data must be summed for monthly aggregation. Selecting the appropriate period for analysis is critical. The aggregation period is relative to the application, but is nonetheless necessary. Figure 7 demonstrates data aggregation depending on the period selected. The time in the example is in calendar days.

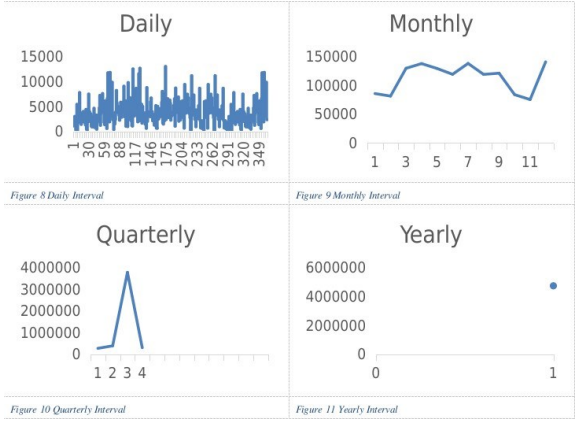


Figure 7: Visualizing periodized data aggregation in time-series data.

Selecting the appropriate interval eliminates fluctuations while keeping the essence of the behavior. Looking at a dailyinterval may show a high fluctuation. Relatedly, the daily interval renders it difficult to interpret if the time-series is too long. Quarterly and yearly provide too little data points to fully capture the essence of the behavior. The patterns tend to appear similar with too much aggregation. Based on this example, the monthly interval gives the best description of behavior.

Further, the sum of absolute difference and sum of series will reveal patterns in the data. Thus, it is preferable to havetoo many data points rather than too few. For example, daily and monthly (Figure 7) have similarities based on the output while the pattern is difficult to see in daily visualizations. If the time points measured are less (e.g., quarterly and yearly), then the scores show similarities between different series that may not be present when visualized monthly. Of note, data over a longitudinal period (e.g., ten years) could yield deeper insight whereas monthly may be too noisy. This highlights the importance of selecting the correct interval based on the application.

4.2 Rounding

Rounding is another way to optimize time-series output. Similar items will still be clustered together based on the score, however rounding allows for a smoothness of the series and leads to more rigid values. In this way, the fluctuation evolves towards a more fluid pattern. To demonstrate (Figure 8), given twelve values between 1000 and 1005, the data were rounded to the nearest thousand. Doing so achieves the goal to smooth fluctuations in relatively similar values.

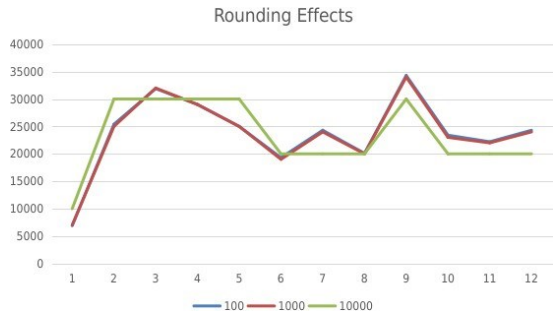


Figure 8: Visualization of smoothed time-series outputs.

Given the data set, when rounded to the nearest hundred and thousand the results are similar and does not add any level smoothness to the series. When rounding to the nearest ten thousand though, the variation becomes much less and causes the series to exhibit less fluctuation. This step can provide more robustness over the model to aid in visualizing patterns but must be completed with caution.

Rounding can cause time-series data, intended to be separated, to have similar scores. A sound approach then is rounding to specific places based on data features (e.g., rounding to the nearest ten for car speeds, hundred for plane speeds, and thousands for rocket speeds). As well, for clarity- rounding aids with visualization. Rounding is not necessary in computing the sum of absolute difference of sum of series.

5. Conclusion

Time-series analysis is a critical tool in a variety of industries. The chief use of time-series in these industries is to compute predictions based on a historical dataset. Such predictive capacity gives society the ability to predict diverse events such as cyberattacks, financial market changes, weather forecasting, and disease outbreaks. Further, clustering is heavily employed to aid in time-series data analysis, particularly when the data have a temporal attribute. However, existing algorithms for analyzing time-series data fail to quantify and visualize how different series compare based on temporal fluctuations [7, 8, 9]. Thus, a gap exists because clustering categorical data by behavioral patterns is computationally expensive and does not provide insight beyond how a single series of data is related to another from a time-series perspective (e.g., a series that has high fluctuation or variation versus low).

Identifying a new way to cluster time-series data was motivated by a need to overcome the existing, limited approaches. This included the creation of the behavior algorithm which computes the sum of absolute difference and sum of series features. Overall, the behavior algorithm demonstrates one potential solution to the research problem.

More specifically, the comparison of the absolute, normalized, and z-score values combined with computing the traditional statistics, sum of absolute difference, and sum of series, identified which values appropriately quantify the behavior of the series. Further, the sum of absolute difference and sum of series of the normalized data was able to distinguish the difference of data, even given the changes in absolute values. In this way, the sum of absolute difference and sum of series values overcome limitations in time-series clustering.

Lastly, providing a single value describing the behavior of the series and improving the computational complexity by eliminating the need to measure every data point (i.e., DTW), or by finding the difference between every time point in one series compared to another (i.e., Euclidean distance). Now with one value, clustering can be done in a fraction of the time, with very reliable results. This is achieved by computing the sum of absolute difference and sum of series for every series, and then either clustering all series together or using the values for classification using machine learningalgorithms.

With respect to future work, we first suggest work be done to produce an operational prototype implementing the time-series behavior algorithm. Of course, validation experiments would be necessary to quantify efficacy of the approach. Additional work then could be done to implement the behavior algorithm in a machine learning pipeline. Ex- perimentation in this area could develop a robust basis for understanding appropriate and inappropriate implementations across types of classifiers (i.e., supervised versus unsupervised). Finally, future work may be of interest in computational optimizing of behavior algorithm. Again, experimentation may reveal space and time complexity optimization with generalizability across types of time-series data.

6. References

[1] Abdulkadir Bilen and Ahmet Bedri Özer. Cyber-attack method and perpetrator prediction using machine learning algorithms. *PeerJ Computer Science*, 7:e475, 2021.

[2] E Ramanujam, Thinagaran Perumal, and S Padmavathi. Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. IEEE Sensors Journal, 21(12):13029–13040, 2021.

[3] Deepak Kumar, Pradeepta Kumar Sarangi, and Rajit Verma. A systematic review of stock market prediction using machine learning and statistical techniques. Materials Today: Proceedings, 2021.

[4] Mohammad Shehab, Laith Abualigah, Qusai Shambour, Muhannad A Abu-Hashem, Mohd Khaled Yousef Shambour, Ahmed Izzat Alsalibi, and Amir H Gandomi. Machine learning in medical applications: A review of state-of-the-art methods. Computers in Biology and Medicine, 145:105458, 2022.

[5] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. Econometric reviews, 29(5-6):594–621, 2010.

[6] D Neha and BM Vidyavathi. A survey on applications of data mining using clustering techniques. International Journal of Computer Applications, 126(2), 2015.

[7] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. International journal of forecasting, 22(3):443–473, 2006.

[8] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine learning strategies for time series forecasting. In European business intelligence summer school, pages 62–77. Springer, 2012.

[9] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. Big Data, 9(1):3–21, 2021.

[10] Chong Li, Yingjie Yang, and Sifeng Liu. A new method to mitigate data fluctuations for time series prediction. Applied Mathematical Modelling, 65:390–407, 2019.

[11] Yang Lin, Irena Koprinska, and Mashud Rana. Springnet: Transformer and spring dtw for time series forecasting. In International Conference on Neural Information Processing, pages 616–628. Springer, 2020.

[12] Galit Shmueli and Kenneth C Lichtendahl Jr. Practical time series forecasting with r: A hands-on guide. Axelrod Schnall Publishers, 2016.

[13] Srihari Athiyarath, Mousumi Paul, and Srivatsa Krishnaswamy. A comparative study and analysis of time series forecasting techniques. SN Computer Science, 1(3):1–7, 2020.

[14] T Warren Liao. Clustering of time series data—a survey. Pattern recognition, 38(11):1857–1874, 2005.

[15] Ali Javed, Byung Suk Lee, and Donna M Rizzo. A benchmark study on time series clustering. Machine Learning with Applications, 1:100001, 2020.

[16] Ali Alqahtani, Mohammed Ali, Xianghua Xie, and Mark W Jones. Deep time-series clustering: A review.
Electronics, 10(23):3001, 2021.

[17] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review.
Information systems, 53:16–38, 2015.

[18] Jorge Caiado, Elizabeth A Maharaj, and Pierpaolo D'Urso. Time series clustering. Handbook of cluster analysis, pages 241–263, 2015.

[19] Ville Hautamaki, Pekka Nykanen, and Pasi Franti. Time-series clustering by approximate prototypes. In 2008 19th International conference on pattern recognition, pages 1–4. IEEE, 2008.

[20] Pavel Senin. Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855(1-23):40, 2008.

[21] Meinard Müller. Dynamic time warping. Information retrieval for music and motion, pages 69–84, 2007.

[22] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(6):623–635, 1980.

[23] Phudit Thanakulkairid, Tanupat Trakulthongchai, Naruesorn Prabpon, and Pat Vatiwutipong. Efficiency of time series clustering method based on distribution of difference using several distances. In 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), pages 1–6, 2022.

[24] Michio Sugeno and Takahiro Yasukawa. A fuzzy-logic-based approach to qualitative modeling. IEEE Transactions on fuzzy systems, 1(1):7–31, 1993.

[25] Ruey S Tsay. Outliers, level shifts, and variance changes in time series. Journal of forecasting, 7(1):1–20, 1988.

[26] Nathan S Balke. Detecting level shifts in time series. Journal of Business & Economic Statistics, 11(1):81–92, 1993.

[27] Nathan S. Balke. Detecting level shifts in time series. Journal of Business Economic Statistics, 11(1):81–92, 1993.

[28] Michael R Berthold and Frank Höppner. On clustering time series using euclidean distance and pearson correlation.
arXiv preprint arXiv:1601.02213, 2016.

Change Management Maturity Model

Henning Lübbecke, Independent Researcher

***Abstract*— Since computer systems exist, a lot of implementation projects fails. The Implementation of computer systems causes changes in processes of work and behavior of people. Change Management is one of the most important success factors, when implementing software systems in organizations. Because it is the key to successfully transform business processes. The ability to change varies between the different organizations. This ability depends on different critical success factors. To evaluate the ability to change of an organization, a maturity model is introduced. Therefore, the critical success factors of change are presented. With this critical success factors as key process areas, the development of the change management maturity model starts. In a five stage maturity model, the shape of each key process area is defined. The model is evaluated by experts for maturity models and experts for change management. The result is a maturity model to assess the ability to change of an organization.**

***Keywords*— change management, computer systems, critical success factors, design science, maturity model.**

I. Introduction

In 1968 the first software engineering conference was held in Garmisch-Partenkirchen. The "software crises" was discussed, because large number of software projects fail [32]. From this starting Point "the trend in successful project delivery seems to be slowly improving from a historical low point of 29% in 2004 to a more recent figure of 39% [37], well over half of IS projects are not delivered successfully" [17]. Today, the Standish Group [37] assumes that 66 % of software projects fail [37]. Why is that so? To answer this question, numerous papers describes success factors for implementing software systems. One of the factors is Change Management. In his paper, Kotter [23] describes, that change management is the key to successfully transform business processes.

The introduction of a new software system is nothing else than a transformation of a business process. The transformation is sometimes successful, sometimes not. Therefore, the success factors of change management were investigated. The results of this investigation are recorded in [28]. To evaluate the ability to transform of an organization, a maturity model was developed. In the next section, a method to develop a maturity model is presented. Afterwards, the critical success factors of change management are introduced. In the main the section the model development process is described, and the model is described.

II. Change Management

The goal of change management in IT-projects is to document, plan, and implement all changes [22]. The holistic and systematic planning, initiating, realizing, reflecting and stabilizing of change processes is the purpose of change management. There are two types of change: transitional and transformational. Transitional change is the crossover from an unsatisfying initial state into a jointly supported final state. Supporting individuals in their own learning process is called transformational change. Both kinds of change are addressed by change management because the implementation of IT-systems needs usually the change of business processes and a change in employee behavior. With change management a better culture and an appreciatively together should be reached [22]. Change management is a continuous improvement process leading to an agile and learning organization [22]. The most popular model of change management is the action research 3-phase-model from Kurt Lewin [10, 22, 23, 24, 20, 39]. Lewis Model defines a process with three phases. First the organization must be "unfreeze". That means established hierarchies and processes are questioned. The next phase is "moving". This phase is the real transformation. Last phase is "freezing". In this phase new processes and a new organization structure is settled. In his bestseller "Leading Change" Kotter [23] transforms Lewis Model in an abstract universal plan for change management [22]. Kotter's Plan consists of eight steps. First step is establishing a sense of urgency. Therefor the market and the existing competitive is examined. Crisis, potential crises, and major opportunities be considered. The next step is the forming of a powerful coalition by assembling a group with enough power to lead the change and encourage is team. Third step is the creation of a vision and strategies to implement this vision. Afterwards the communication of the vision is the next step. The following step is to empower others to act on the vision. Planning and creating short-term wins follows. The seventh step is consolidation improvements and producing more change. The last step is institutionalizing the new approach.

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

The next few paragraphs should contain the authors' current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@ boulder.nist.gov).

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu).

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan (e-mail: author@nrim.go.jp).

III. Maturity Models

Maturity models are stage models which describes an evolution path. They can be used for comparing and/or for continuous development. They arise from quality management and offer benchmarking and performance improvement [14]. The most prominent maturity model is the CMM or rather CMMI from Software Engineering Institute (SEI) of the Carnegie Mellon University. With this model maturity models getting popular and variety of models for a lot of issues are developed.

Becker et al. [8] developed a method to develop maturity models. They use a design science approach to develop this method. First, they derive eight requirements from Hefner's Guidelines for Design Science for the development of maturity models:

1. Comparing with other maturity models to justify the development of a new maturity model,
2. Iterative approach,
3. Evaluation of usefulness, quality and effectivity of the new maturity model,
4. Multimethod approach,
5. Relevance of the problem,
6. Definition of the issue,
7. Presentation of the results appropriate to the addressee,
8. Scientific documentation.

Becker et al. [8] start with the definition and relevance of the issue (requirements 5 and 6). This can be done with a literature review. With this literature review also the comparison with other maturity models can be done. Is there no such model, that solves the issue, the need to develop a model exists. Otherwise, the existing model is to be checked whether it solves the issue or should be modified. Is there no model or is it necessary to modify an existing model, then developing of the development strategy starts. The development process of the maturity model or model modification is a Demming-Circle which is to be run through

Iteratively. The steps in the circle are: choice of the design level, choice of the approach, model design, and test. This circle is being run from the architecture of the model over a variety of (part-)model on different levels of abstraction until the model is created. With the documentation of the different approaches and on the different design levels and with the test the requirements 2, 3 and 4 are fulfilled. After the iterative development of the model, the transfer and evaluation of the model follow to accomplish the requirements 3 and 7. Becker et al. [8] envisage the continuous development of the model until its retiring.

IV. Development of the Change Management Maturity Model

According to the method of Becker et al. [8] a literature review was done. The literature reviews follows the method of Kitchenham and Charters [21]. Two literature reviews were done. One literature review on the success factors of change management. This literature review is documented in [28). This review shows the relevance of the problem. The second literature review asks about change management maturity models. With this two literature reviews, the requirement 1 (comparing with other models) is fulfilled.

A. Problem definition and relevance

The Standish Group [37] assumes that 66 % of software projects fail [37]. One of the main critical success factors for the success or flop of a software implementation is change management [22]. Thus the ability of an organization to change is relevant and to estimate the prospect of success of a project. The change management maturity of the organization is therefore important. With this the requirements 5 and 6 are done.

B. Literature review on critical success factors of change management in software projects

Lübbecke [28] found with a literature review 22 critical success factors for change management in software projects. These factors are: stakeholder involvement, culture, communication, training, top management support, team work, leadership, vision, testing, technical infrastructure, champions, resource availability, audit, incentive system, team efficiency, feedback, cross-departmental cooperation, project management, partnership, knowledge management, process reengineering, and the organization structure. The literature review is documented in [28].

C. Literature review on change management maturity models

The systematical literature review starts with identification of reasons and goals. In this case, it is to find a maturity model for change management. After this, protocol and evaluation of the literature review are designed. According to Kitchenham and Charters [21] the databases IEEExplore, IET Digital Library, Google Scholar, Citeseer library, Inspec, and ScienceDirect were searched. Search String was „change management maturity model". The result was: 849,000 hits at Google Scholar, 153 hits at IEEExplore, 1.477.998 hits at Citeseer Library, 238 hits at ScienceDirect, 48,152 hits in Inspec, and 1695 hits in IET Digital Library.

The hits were sorted by relevance, and the first 100 hits per database were analyzed by headlines and abstracts. 105 papers describes a maturity model.

No paper describes a maturity model for change management. Next Step is the development of the model in a iterative, multimethod approach.

D. Iterative, multimethod approach

Because there is no model, the development strategy is designed. The development is a Demming-Circle with the steps choice of the design level, choice of the approach, model design, and test. From model architecture over a variety of (part-) models on different abstraction layers, the circle is run through until the model is ready. It is an iterative process, the documentation of the diverse abstraction layer, approaches and tests to fulfil requirements 2, 3, and 4.

After the model development, it is necessary to design concepts for transfer and the evaluation of the model. These concepts include a description of how to realize requirements

7 and 8. Afterwards, implementation of the model transfer and the evaluation of the model takes place.

1) First iteration

Lübbecke [28] found with a literature review 22 critical success factors for change management in software projects. These factors are: stakeholder involvement, culture, communication, training, top management support, team work, leadership, vision, testing, technical infrastructure, champions, resource availability, audit, incentive system, team efficiency, feedback, cross-departmental cooperation, project management, partnership, knowledge management, process re-engineering, and the organization structure. The literature review is documented in [28].

The success factor were defined as key process areas of the maturity model. With this the first layer is modelled.

2) Second iteration

For each key process area, a definition was given in the form of a description. Therefore, a literature search for each success factor was done. On the result of this search, a descriptive definition for each key process area was created. So the second layer of the model is created.

3) Third iteration

With literature reviews, maturity models for each key process area were searched. The different models for each key process area were compared and combined. The third layer is ready.

4) Forth iteration

The different maturity models were integrated and consolidated into one maturity model with five stages. This is the first version of the complete model.

V. Evaluation of the Model

To classify the different evaluation possibilities, Helgesson et al. [16] introduced the following framework. It consists of three evaluation types. First, evaluation by the author on basis of his expertise. Second, evaluation by practitioners with surveys, interviews or assessments. Third, evaluation by practice. The three types of evaluation suggest a chronically order of evaluations. Beginning with the evaluation of the model by model experts, normally the author, then by domain experts, usually practitioners, and at least by use in practice.

Regard to the here presented model, the first step of the evaluation was done by the author in form of proofreading.

For the second evaluation step, Salah and Cairns [34] developed a questionnaire. This questionnaire was translated into German and send to experts for change management. 7 experts responded. The answers and remarks were analyzed and in the model integrated respectively the model was changed.

When reworking the model the criterion champion is integrated in leadership, team work and team efficiency are conflated just as stakeholder involvement and partnership. So there was a reduction from 22 success factors to 19 key process areas.

VI. The Model

The model is a five stage model with 19 key process areas. The stages are:

1. initial, no requirements,
2. managed, projects are planned and documented,
3. defined, there is a standard process,
4. quantitatively managed, there is a controlled standard process, and
5. optimized, there is a continuous improvement process.

The key process areas are stakeholder involvement, culture, communication, training, top management support, team, leadership, vision, test, technical infrastructure, resource availability, audit, incentive system, feedback, cross-departmental cooperation, project management, knowledge management, process reengineering, and the organization structure.

Stakeholder of a project are shareholder, owner, management, clients, customer, supplier, and employees. Management of expectations, understanding of stakeholder requirements, fostering the system understanding of the stakeholder, increasing system acceptance, overcoming resistances, and preparation of change are the topics of stakeholder involvement [6, 29].

Culture is a set of ideas, assumptions, beliefs, standards, principles, behaviours, and values like responsibility, customer orientation, willingness to change, teamwork and nature of decision-making [2, 3, 7, 9, 11, 12, Müller et. al., 2009, Urban, 2015).

Generation, dissemination, configuration, transfer of information and the interaction in a social environment are the topics of communication [13, 24].

Training is precondition to adopt new modes of work and behaviour.

Top management support is an important factor for changes, it helps to overcome resistance, provides technical, financial and human resources, takes leadership, has strategies, serves as role model, makes decisions and has the responsibility, represents the changes, is member of the steering board, gives incentives and rewards, is part of assessments and workshops [6, 36, 38, 41, 42, 44, 45].

The team needs the right skills and qualifications in sufficient extent qualitative and quantitative. The team members must be able to work together.

To reach new behaviour and business processes leadership is necessary.

A goals defining vision is the starting point for every change project [7, 26, 41, 42].

The development process, testing and troubleshooting are the way to successful information system implementation [6, 42].

A sufficient technical infrastructure is the key basic for the operation of an information system.

For the success of every project sufficient resources are necessary [5].

To control the project progress and the quality of created artefacts, audits takes place [18, 30, 40, 43].

Behaviour changes and the collaboration in and with change projects can by motivated by incentives [13, 19].

Self-efficacy and self-control will be increased by feedback, also system and/or process design can be improved by feedback [1, 43].

To implement new behaviour and processes cross-departmental cooperation is inevitable [42].

Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements [27].

Every organization has internal and external partners who are involved or affected by the change project.

Knowledge management is the set of all activities to collect, organize, manage, distil, present, share, use, and store knowledge [15].

Process improvement is the issue of business process re-engineering [31].

The organisation structure is described by its hierarchy and procedures [33].

A. Maturity level 1

On this level there is no stakeholder involvement in change projects. Partnerships are fix, there is no cooperation with regard to the intended change.

A welcoming culture did not exist in the organization. There is no induction concept for new employees.

Employees are not encouraged to produce innovations and ideas, therefore they did not communicate their ideas. There is only poor communication between Employees and between departments. A communication culture did not exist. Communication is informal, there is no formal reporting or status meetings. The content of change projects is not communicated.

The knowledge in the organization about change is not sufficient to design an appropriate training plan. The top management support is missing. Team work did not happen. The success depends on individuals. Work is function oriented. Work in teams is not supported by the management. There is command but no leadership. The link between management vision and intended change is not discernible.

Test is chaotic and takes place with debugging. Test cases design is ad hoc. The result is software without big bugs. The organization has no effective control on its infrastructure. The processes are not managed. Standards, guidelines and procedures are missing. Resources are not planned, a systematic overview on the availability of resources is missing. Audits are unknown. There is no incentive system.

Decision-making in the organization is informal, individual and weak defined. It is based on personal, tacit experiences and individual appraisal. Decisions are not documented. There is no feedback on the efficiency of a decision.

Cooperation ends at the frontier of the own department.

There is no defined project management process. Project processes are not documented. Documentation standards did not exist.

Knowledge management practical did not exist. It is not defined. A concept is not available in the organization. Processes of knowledge management are operated unconscious, unsystematic, and aimless.

Processes are not structured and bad defined. Process measurement did not exist and workplace and organization structure are based on traditional functions, and not on horizontal processes.

Local and functional layers are autonomous. There are no organization wide standards, governance, and formal ratings.

Goals are not named. Therefore, the goal achievement is not rated. There is no governance. Processes are unknown, not documented, not measured and not controlled. A conscious examination with the organization structure and the used technology did not happen.

B. Maturity level 2

At maturity level 2 activities are planned, controlled, and documented. To enhance the involvement of the employees, information about changes and the progress of the project is given to them. Only internal Partnerships are on the focus, there is still no involvement of external partners. The organization encourages ideas and innovation and takes initiative to experiment with new concepts. Employees feel involved in decision processes. The organization recognize employees as valuable resources and their doing as important for successes. There is a communication protocol. Employees are informed about changes and the progress of the change project. Informal reports exist and reviews are implemented. Information systems occur. Training is offered. Employees are able to discuss knowledge and experiences with each other. The top management take notice of the change and received reports on the project.

Teamwork is focused, it takes place occasionally, and it is atypical. Each team has its own, individual management and controlling approach.

The top management knows the necessity of a performance improvement. One member of the top management supports the operational changes and provides resources. The implementation is done by the middle management. The change is part of the vision of the top management. The organization has an IT strategy that implement the operational requirements of the organization.

Test is an independent process that takes place after implementation. Introduction of test plans and test strategies starts. Stakeholders take part. Risk management techniques are used to identify product risks. The organization controlled their key infrastructure. For fundamentals, guidelines exist. There are enough resources available. A poor audit trace exists and can by noticed.

An incentive system is defined. Decision-making is informal. Experiences and opinions that lead to the decision are documented. There is no feedback on the efficiency of decisions.

Cooperation between departments is like „corresponding silos“. It happens in a clearly-outlined on few organizational interfaces.

Project management is weakly used, the process and the documentation are informal. Positive decisions on projects to implement knowledge management exist. Knowledge artifacts are identified. First structures are implemented. The essential

processes are identified and documented. Work places and organization structures include process elements. Representatives of functions meet regularly for coordination. First organization wide standards or governance exist, but no formal measurement system. Goals are defined, but not measured. The necessity of governance is known. Processes are named and assessments run. There is an open discussion on. The implementation of change management and leadership runs. The used technology is identified.

C. Maturity level 3

This maturity level is the level where processes starts. At maturity level 2 the activities are planned, controlled, and documented, now the activities are defined and standardized. Though processes oriented change management starts.

The staff is involved in change projects. Internal partnership are established, customer and/or supplier are involved if necessary. The staff understand the vision, goals and values, and agree to them. Employees decide how to do tasks. Information about change, its strategical alignment, and the plans about implementation are known beyond department boundaries. All involved and affected people know the scope of the change. For documentation, structured data formats are used. Analyses, reviews and assessments occur regular. Necessary trainings are offered. Employees are able to participate by problem-solving and idea generation activities. Information from external resources are available for all. Formal and informal, internal and external discussion forums are open to all. Employees are encouraged to experiment with their knowledge.

The top management is present in the project. Interdisciplinary teams work on change projects. Norms and standards, expert knowledge and tools are used. There are informal project management trainings. Customer and suppliers are involved. A member of the top management took over the role of the leader, declares goals and took responsibility for the program. The change strategy is aligned with die business and IT strategy and documented.

Test is integrated in the development process. Test planning takes place in early phases. There is a standard test process, a test organization unit, trainings on tests are provided, and a formal review program exists. There is a standard infrastructure with SLA's and a proactive problem management. Human and financial resources are available. Audits are documented. An incentive system is established. Decision-making is formalized and accepted in the organization. Basics and results are transparent. There is no feedback on the efficiency of the decision. In phases, cross-departmental cooperation respectively assistance takes place.

The project management process with controlling, formal planning, and data collection is performed consistently. Knowledge and experiences are identified. The organization structure with roles and responsibilities for knowledge management are defined. Standards, policies, tools and methods are documented. A training program is provided. The business process is the basic of the organization. Work places are aligned to processes. Process metrics measure the performance. Process management is used to reach strategy goals. Goals are defined, communicated and measured. Work flows are designed. Risk management is established. Change management and leadership is practiced. The used technology is designed.

D. Maturity level 4

When at maturity level 3 processes where implemented, the measurement of this processes is the main topic at level 4. The goal of this maturity level is to manage the change process quantitatively.

Forums and teams for exchange are established. Long-time partnerships with customer and suppliers are arranged. The management promote changes. The staff understands changes, wage it positive, and supports it. Interdisciplinary teams pass, share, and keeps knowledge. Open, cross-departmental communications takes place. The whole staff has all necessary information. Communication protocols for communication with external partners exist. Communication technique and communication process are implemented. Formal and informal mechanism of learning and knowledge transfer are used. The organization learns from its experiences and prevent the repetition of failures. Innovation is aligned to the goals of the organization. The top management is active, engaged in changes. Teamwork is mandatory. The top management works as a team, it controls the organization by processes, has a vision and delegates the controlling to the process owners. More and more others in the organization take over the role of leaders. The realization of the vision is measured by processes.

There is a technical and personnel infrastructure. The test process include inspections in the development process. The test is continuously evaluated by a measurement program. Infrastructure and their processes are optimized. Continuous service advancement take place, SLAs are flexible. Technical, human and financial resources are provided. The audit process is controlled and on improvement analyzed.

The incentive system is operated. Decision-making is formalized and accepted in the organization. Basics, process and results are transparent. There is no feedback on the efficiency of decisions. There is cross-departmental cooperation in projects. The project management process is mature. The project management process is measured and quantitative analyzed. The results of measurement and analyses are the basis for process improvement. Build, share, and use of knowledge is organizational integrated and quantitative evaluated. Employees expect to be successful when searching for knowledge artifacts. Documentation of knowledge management tools is available. Organization, customer, and supplier cooperate on basic of processes. Processes are the basic of organization structure and work places. Process measurement and management systems are implemented in the organization. The measurement of goal attainment ensued continuously by metrics. Work flows are implemented and accepted. Change management is used strategically

E. Maturity level 5

Now the change process is defined and measured. The results of the enduring measurement are basic for continuously improvement of the change process.

There is a complete, and optimized supply-chain. A conflict resolution plan is established. All members of the organization value quality, service and reliability. Innovation are supported. Communication is in free flow and accurate recognized. There is a system to optimize communication. Research and development are continuous processes in the organization. The organization learns. The Top Management is active engaged in and for the project. Teamwork with customers and suppliers is mandatory. People out of the whole organization took over leadership and get qualification therefore. Strategical planning and policies are regularly updated and consider information to performance, customer, supplier, and other facilities that interrelate with the organization. Benchmarks are operated.

The test process is optimized. It is quantitative measured and is continuously improved. The infrastructure is robust and agile. It is a catalyst for innovation. The resource provision is demand-driven. The audit process is optimized.

The incentive system is optimized. Decision-making is formalized and accepted in the organization. Basics, processes and results are transparent. Feedback on the efficiency of decisions exists and is documented. It is the basis for future decisions. Cross-departmental cooperation is mandatory. The project management organization is mature, complete understood, and continuously optimized. Project management is integrated into the management. The organization uses its extensive knowledge to optimize continuously processes. The process integration inclusive partners is completed. Continuous adaptation to market requirements takes place. Goals are continuously evolved, the vision is constantly aligned. Results are evaluated and processes evolved.

VII. Conclusion

With a design science approach, a maturity model was designed and evaluated. The evaluation was carried out by change management experts. This model consists of 22 key process areas. The model measured the maturity of organizations to change, when implementing software system. With the results the organization have the possibilities to change in factors relevant for changes to have a greater likelihood to be successful in information system implementation.

References

[1] Agapie, E., Avrahami, D., & Marlow, J. (2016, May). Staying the course: system-driven lapse management for supporting behaviour change. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 1072-1083). ACM.

[2] Ahmed, F., & Capretz, L. F. (2010). An organizational maturity model of software product line engineering. *Software Quality Journal*, *18*(2), 195-225.

[3] Alrasheedi, M., Capretz, L. F., & Raza, A. (2016). Management's perspective on critical success factors affecting mobile learning in higher education institutions—An empirical study. *Journal of Educational Computing Research*, *54*(2), 253-274.

[4] Alreemy, Z., Chang, V., Walters, R., & Wills, G. (2016). Critical success factors

[5] Alshibly, H., Chiong, R., & Bao, Y. (2016). Investigating the critical success factors for implementing electronic document management systems in governments: evidence from Jordan. *Information Systems Management*, *33*(4), 287-301.

[6] Altamony, H., Al-Salti, Z., Gharaibeh, A., & Elyas, T. (2016). The relationship between change management strategy and successful enterprise resource planning (ERP) implementations: A theoretical perspective. *International Journal of Business Management and Economic Research*, *7*(4), 690-703.

[7] Association of Change Management Professionals (ACMP) (2014). Standard of Change Management.

[8] Becker, J., Knackstedt, R., & Pöppelbuß, J. (2009). Developing maturity models for IT management. *Business & Information Systems Engineering*, *1*(3), 213-222.

[9] Burnes*, B. (2004). Kurt Lewin and complexity theories: back to the future?. Journal of change management, 4(4), 309-325.

[10] Cooke-Davies, T. J., & Arzymanow, A. (2003). The maturity of project management in different industries: An investigation into variations between project management models. *International Journal of Project Management*, *21*(6), 471-478.

[11] Corriss, L. (2010, December). Information security governance: Integrating security into the organizational culture. In *Proceedings of the 2010 Workshop on Governance of Technology, Information and Policies* (pp. 35-41). ACM.

[12] Cosic, R., Shanks, G., & Maynard, S. (2012, January). Towards a business analytics capability maturity model. In *ACIS 2012: Location, Location, Location: Proceedings of the 23rd Australasian Conference on Information Systems 2012* (pp. 1-11). ACIS.

[13] Curtis, B., Hefley, B., & Miller, S. (2009). *People capability maturity model (P-CMM) version 2.0* (No. CMU/SEI-2009-TR-003). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

[14] Demir, C., & Kocabaş, İ. (2010). Project management maturity model (PMMM) in educational organizations. *Procedia-Social and Behavioral Sciences*, *9*, 1641-1645.

[15] Gunjal, B. (2019). Knowledge management: why do we need it for corporates. *Malaysian Journal of Library & Information Science (ISSN: 1394-6234)*.

[16] Helgesson, Y. Y. L., Höst, M., & Weyns, K. (2012). A review of methods for evaluation of maturity models for process improvement. *Journal of Software: Evolution and Process*, *24*(4), 436-454.

[17] Hughes, D. L., Rana, N. P., & Simintiras, A. C. (2017). The changing landscape of IS project failure: an examination of the key factors. *Journal of Enterprise Information Management*.

[18] IEEE Std. 610, "IEEE Std 610: Standard Computer Dictionary", Institute of Electrical and Electronic Engineers, USA, 1991.

[19] Johansson, E. (2002). Quality Improvement in Software Platform Development.

[20] Kaminski, J. (2011). Theory applied to informatics–Lewin's Change Theory. *Canadian Journal of Nursing Informatics*, *6*(1).

[21] Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

[22] Kostka, C. (2016). Change Management: Das Praxisbuch für Führungskräfte. Carl Hanser Verlag GmbH Co KG.

[23] Kotter, J. P. (1995). Leading change: Why transformation efforts fail.

[24] Kwak, Y. H., & Ibbs, C. W. (2002). Project management process maturity (PM) 2 model. *Journal of management in engineering*, *18*(3), 150-155.

[25] Lewin, K. (2016). Frontiers in group dynamics: Concept, method and reality in social science; social equilibria and social change. Human relations.

[26] Litzcke, S. M., & Nolte, R. (2008). *Change Management-Theorie und Praxis*.

[27] Lückmann, P., & Färber, K. (2016). The impact of cultural differences on project stakeholder engagement: a review of case study research in international project management. *Procedia Computer Science*, *100*, 85-94.

[28] Lübbecke, H. (2021). Critical Success Factors of Change Management in Software Projects. *Вестник Томского государственного университета. Экономика*, (55), 185-197.

[29] Niazi, M., Wilson, D., & Zowghi, D. (2005). A maturity model for the implementation of software process improvement: an empirical study. *Journal of systems and software*, *74*(2), 155-172.
[30] Paulk, M. C., Weber, C. V., Garcia, S. M., Chrissis, M. B. C., & Bush, M. (1993). Key practices of the capability maturity model version 1.1.
[31] Qu, Y., Ming, X., Ni, Y., Li, X., Liu, Z., Zhang, X., & Xie, L. (2019). An integrated framework of enterprise information systems in smart manufacturing system via business process reengineering. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 233(11), 2210-2224.
[32] Randell, B. (2018). Fifty years of software engineering-or-the view from garmisch. *arXiv preprint arXiv:1805.02742*.
[33] Rowlinson, S., & Cheung, Y. K. F. (2008). Stakeholder management through empowerment: modelling project success. *Construction Management and Economics*, *26*(6), 611-623.
[34] Salah, D., Paige, R., & Cairns, P. (2014, December). An evaluation template for expert review of maturity models. In *International Conference on Product-Focused Software Process Improvement* (pp. 318-321). Springer, Cham.
[35] Shirish, A., & Batuekueno, L. (2021). Technology renewal, user resistance, user adoption: status quo bias theory revisited. *Journal of Organizational Change Management*.
[36] Spoelstra, W., Iacob, M., & van Sinderen, M. (2011, March). Software reuse in agile development organizations: a conceptual management tool. In *Proceedings of the 2011 ACM Symposium on Applied Computing* (pp. 315-322). ACM.
[37] Standish Group. (2013). Chaos manifesto 2013. *The Standish Group International*.
[38] Stelzer, D., & Mellis, W. (1998). Success factors of organizational change in software process improvement. *Software Process: Improvement and Practice*, *4*(4), 227-250.
[39] Sarayreh, B. H., Khudair, H., & Barakat, E. A. (2013). Comparative study: The Kurt Lewin of change management. *International Journal of Computer and Information Technology*, *2*(4), 626-629.
[40] Sun, M., Vidalakis, C., & Oza, T. (2009, September). A change management maturity model for construction projects. In *Proceedings 25th Annual ARCOM Conference, of Association of researchers in construction management* (pp. 803-812).
[41] Tabak, R. (2016). Critical success factors for spend analytics.
[42] Totla, K., Mandot, M., & Gaur, S. (2016). An insight of critical success factors for ERP model. *International Journal of Emerging Research in Management & Technology*, *5*, 90-92.
[43] Van Veenendaal, E., & Wells, B. (2012). Test Maturity Model integration TMMi. *Uitgeverij Tutein Nolthenius*.
[44] Yeo, K. T., & Ren, Y. (2009). Risk management capability maturity model for complex product systems (CoPS) projects. *Systems Engineering*, *12*(4), 275-294.
[45] Yeoh, W., & Popovič, A. (2016). Extending the understanding of critical success factors for implementing business intelligence systems. *Journal of the Association for Information Science and Technology*, *67*(1), 134-147.

Prediction of Rainfall based on Statistical and Computational Approach

Writaparna Mukherjee
Institute of Engineering & Management, Kolkata. India
writaparnam@iemcal.com

Angshuman Ray
Institute of Engineering & Management, Kolkata. India
angshuman@iemcal.com

Bimal Datta
The Budge Budge Institute of Technology, Budge Budge. India
bkd_hetc@yahoo.co.in

Pintu Pal
Asansol Engineering College, Asansol. India
p5_pal@yahoo.co.in

***Abstract*— A comparative study is done in this paper in the prediction of rainfall at ground level multiple linear regression and, feature selection and k-means clustering method. Based on the past observations of the last three days atmospheric parameters like minimum and maximum Temperature, minimum and maximum relative humidity, minimum and maximum air pressure, minimum and maximum vapour pressure and minimum and maximum radiation the model is developed. In this paper it is observed that considering the seasonality effect better results can be achieved. It has also been observed that the selection of appropriate features can also improve the performance of the prediction.**

Keywords*□ *Multiple linear regression, feature selection, k-means clustering, rainfall.

I. INTRODUCTION

The turbulent idea of the environment, the gigantic computational force needed to settle the conditions that depict the climate including different boundaries like most extreme temperature, least temperature, relative mugginess, fume pressure, wind speed and bearing, precipitation fall, and so forth which are difficult to count and quantify precisely. Many important initiatives to solve the challenge of weather forecasting using statistical modelling, including machine learning techniques, have been published in the last decade.

In temperature gauging one needs to recognize the occasions the gauge goes on, for instance temperature one hour ahead or least and greatest temperature of a given day. A few works has been done and diverse statistical and computational models have been tried. Hiyam Abobaker Yousif Ahmed and Sondos W. A. Mohamed portray a model of rainfall prediction using temperature, wind speed, and dew point for Khartoum state. The efficiency of the model has been measured by comparing the average value of the mean square error of the training data with the test data. Barrera-Animas and Ari Yair depicted a model which predict rainfall an hourly basis using time series data. Isabelle Roesch and Tobias Günther designed a comprehensive and interactive system that allows users to study the output of recurrent neural networks on both the complete training data and testing data. They follow a coarse-to-fine strategy, providing overviews of annual, monthly and daily patterns in the time series and directly support a comparison of different hyper-parameter settings. They applied our method to a recurrent convolutional neural network that was trained and tested on 25 years of climate data to forecast meteorological attributes, such as temperature, pressure and wind velocity.

II. DATA PREPARATION

The current investigation built up a conglomerate model dependent on the past perceptions of a few meteorological boundaries like temperature, humidity, vapour, radiation etc. as a contribution for preparing the model. The information was gathered day by day by the meteorological division of Dumdum Airport. The parameters of the data acquisition are:

i. Minimum Temperature (Min. Temp(t))
ii. Maximum Temperature (Max. Temp(t))
iii. Minimum Relative Humidity (Min. RH(t))
iv. Maximum Relative Humidity (Max. RH(t))
v. Minimum Air Pressure (Min. Press.(t))
vi. Maximum Air Pressure (Max. Press.(t))
vii. Minimum Vapour Pressure (Min. VP(t))
viii. Maximum Vapour Pressure (Max. VP(t))
ix. Minimum Radiation (Min. Rad.(rd))
x. Maximum Radiation (Max. Rad.(rd))
xi. Rainfall (Rain(r))

The prediction was on rainfall on the basis of past three days atmospheric parameters. This information is stored in an input file. The file contains data of seven years. So, there is an observation of 10 variables three consecutive days, say *t*. In the model the rainfall for the next (t_{th}) day is determined by the atmospheric parameters for the current day i.e. day *(t-3)*. To empower the determination of the best model, the preparation informational collection should cover precipitation at various seasons. So the information for whole years were picked as the preparation informational collections. The data is pre-processed before training. Our model has the following functional relations.

Rainfall(r) = f(Temp_Min(t-1), Temp_Max(t-1), Hum_Mint(t-1), Hum_Max(t-1), Pressure_Min(t-1), Pressure_Max(t-1), Vapor_Min(t-1), Vapor_Max(t-1), Rad_Min(t-1), Rad_Max(t-1), Rainfall(t-1), Temp_Min(t-2), Temp_Max(t-2), Hum_Mint(t-2), Hum_Max(t-2), Pressure_Min(t-2), Pressure_Max(t-2), Vapor_Min(t-2), Vapor_Max(t-2), Rad_Min(t-2), Rad_Max(t-2), Rainfall(t-2), Temp_Min(t-3), Temp_Max(t-3), Hum_Mint(t-3), Hum_Max(t-3), Pressure_Min(t-3), Pressure_Max(t-3), Vapor_Min(t-3), Vapor_Max(t-3), Rad_Min(t-3), Rad_Max(t-3), Rainfall(t-3)) (1)

III. METHODOLOGY

The target of regression analysis is to choose the assessments of limits for a limit that cause the ability to best fit a lot of data insights that you give. In linear regression, the capacity is a linear (straight-line) equation. Likewise with connection, regression is utilized to analyze the connection between two continuous (scale) factors. Nonetheless, regression is more qualified for studying practical conditions between factors. The term useful dependency infers that X partially determines the level of Y.

Furthermore, regression is more qualified than relationship for studying tests in which the specialist fixes the appropriation of X. Regression analysis is utilized to anticipate a consistent ward variable from various autonomous factors. On the off chance that the reliant variable is dichotomous, calculated regression ought to be utilized. (If the split between the two levels of the dependent variable is almost half, then, at that point both determined and direct regression will end up giving relative results). The free factors used in regression can be either constant or dichotomous. Autonomous factors with beyond what two levels can likewise be utilized in regression analyses, however they initially should be changed over into factors that have only two levels. This is called dummy coding and will be talked about later. As a rule, regression analysis is used with normally happening factors, as opposed to tentatively controlled variables, regardless of the way that you can use regression with tentatively controlled components. One highlight remember with regression analysis is that causal connections among the factors can't be resolved. While the terminology is to such an extent that we say that X "predicts" Y, we can't say that X "causes" Y.

In this paper the different independent variables we have considered as

i. Minimum Temperature (Temp_Min(t))
ii. Maximum Temperature (Temp_Max(t))
iii. Minimum Relative Humidity (Hum_Mint(t))
iv. Maximum Relative Humidity (Hum_Max(t))
v. Minimum Air Pressure (Pressure_Min(t))
vi. Maximum Air Pressure (Pressure_Max(t))
vii. Minimum Vapour Pressure (Vapor_Min(t))
viii. Maximum Vapour Pressure (Vapor_Max(t))
ix. Minimum Radiation (Rad_Min(t))
x. Maximum Radiation (Rad_Max(t))

And the dependent variable is Rainfall (Rain(r)).

After the initial regression were performed in the normalized data set, the feature selection algorithm was applied in it. The appropriate features were only kept and the data set were cleaned on the basis of the same. After applying feature selection k-means clustering was applied in the data set. As expected there were 4 significant clusters were the output of the application as the data was collected season wise. Then again the multiple linear regression was performed in each cluster and a significant improved results were found.

IV. RESULT AND OBSERVATIONS

After the input file is prepared, the training is done taking into consideration all the parameters. After the training process is over, multiple linear regression was performed on the collected data set. The result is shown below.

TABLE I. OUTPUT OF MULTIPLE LINEAR REGRESSION ON THE GIVEN DATA SET

Coefficients	Estimate	Std. Error
Regression Constant	675.674930	112.833767
Pressure_Max	0.083326	0.304744
Pressure_Min	0.302526	0.288641
Vapor_Max	-0.05599	0.200713
Vapor_Min	-0.14914	0.185311
Hum_Max	-0.01152	0.051431
Hum_Min	0.057332	0.061827
Temp_Max	0.085435	0.233911
Temp_Min	0.449337	0.243745
Rad_Max	0.320937	0.698987
Rad_Min	0.281948	0.981226
Rainfall_1	0.012865	0.026139
Pressure_Max_2	0.07553	0.315234
Pressure_Min_2	-0.14694	0.32006
Vapor_Max_2	0.557232	0.199978
Vapor_Min_2	0.106149	0.185662
Hum_Max_2	-0.1728	0.051868
Hum_Min_2	-0.06442	0.06022
Temp_Max_2	0.068554	0.252792
Temp_Min_2	-0.35544	0.274286
Rad_Max_2	0.380665	0.720005
Rad_Min_2	0.913103	0.987394
Rainfall_2	-0.00706	0.026926
Pressure_Max_3	-0.97165	0.276571
Pressure_Min_3	-0.01137	0.30947
Vapor_Max_3	-0.5605	0.199342
Vapor_Min_3	-0.36748	0.186541
Hum_Max_3	0.266316	0.053075
Hum_Min_3	0.158933	0.060562
Temp_Max_3	-0.20494	0.253389
Temp_Min_3	0.258418	0.254571
Rad_Max_3	-2.84136	0.673489
Rad_Min_3	-2.04699	0.987444
Rainfall_3	0.058032	0.025154

The output of multiple linear regression gives a linear equati on, which is mentioned below:

Rainfall = (Pressure_Max) * (0.083326) + (Pressure_Min) * (0.302526) + (Vapor_Max) * (-0.05599) + (Vapor_Min) * (-0.14914) + (Hum_Max) * (0.01152) + (Hum_Min) * (0.057332)+Temp_Max) * (0.085435) + (Temp_Min) * (0.449337) + (Rad_Max) * (0.320937) + (Rad_Min) * (0.281948) + (Rainfall_1) * (0.012865) + (Pressure_Max_2) * (0.07553) + (Pressure_Min_2) * (0.14694) +

(Vapor_Max_2) * (0.557232) + (Vapor_Min_2) * (0.106149) + (Hum_Max_2) * (-0.1728) + (Hum_Min_2)* (0.06442) + (Temp_Max_2) * (0.068554) + (Temp_Min_2) * (-0.35544) + (Rad_Max_2) * (0.380665) + (Rad_Min_2) * (0.913103) + (Rainfall_2) * (-0.00706) + (Pressure_Max_3) * (-0.97165) + (Pressure_Min_3) * (-0.01137) + (Vapor_Max_3) *(-0.5605) + (Vapor_Min_3) * (-0.36748)+ (Hum_Max_3)* (0.266316) + (Hum_Min_3)* (0.158933) + (Temp_Max_3) * (-0.20494) + (Temp_Min_3)* (0.258418) + (Rad_Max_3) * (-2.84136) + (Rad_Min_3) * (-2.04699) + (Rainfall_3) * (0.058032) +(675.67493) (2)

It was recorded that Residual standard error is 11.54 on 140 0 degrees of freedom and Multiple R-squared value is 0.231 9, Adjusted R-squared value is 0.2138.The root mean square d error is 15.83106, which is significantly high.

To achieve more accuracy of the model, feature selection algorithm was applied in the data set to reduce the number of variable of the data set to reduce the computational cost of modelling. Here the Boruta package were used to perform feature selection in R programming and 500 iterations were performed.

Boruta is a feature selection algorithm. Precisely, it works as a wrapper algorithm around Random Forest. This package derive its name from a demon in Slavic mythology who dwelled in pine forests. This technique achieves supreme importance when a data set comprised of several variables is given for model building. Below is the step wise working of boruta algorithm:

1. Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).
2. Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.
3. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z score than the maximum Z score of its shadow features) and constantly removes features which are deemed highly unimportant.
4. Finally, the algorithm stops either when all features gets confirmed or rejected or it reaches a specified limit of random forest runs.

Out of 33 parameters 28 attributes were confirmed important, 3 attributes confirmed unimportant: Rad_Max, Rad_Max_2, Rainfall_1 and 2 tentative attributes left: Rad_Min_2, Rainfall_2.The results are shown below.

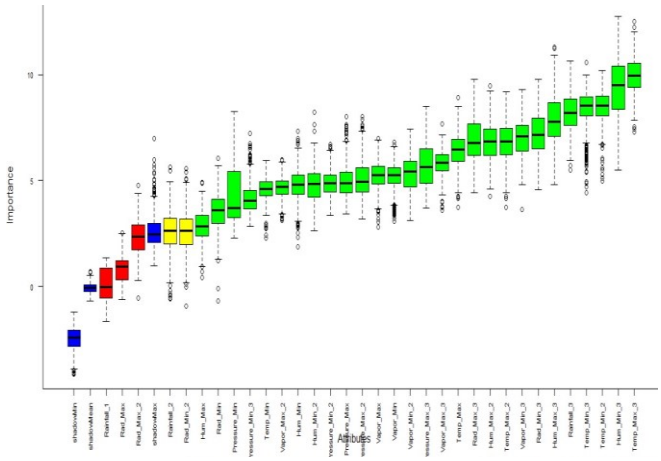


Fig. 1. Correlation of the features of the data set

TABLE II. BORUTA FINAL DECISION

Features	Boruta final decision
Pressure_Max	confirmed
Pressure_Min	confirmed
Vapor_Max	confirmed
Vapor_Min	confirmed
Hum_Max	confirmed
Hum_Min	confirmed
Temp_Max	confirmed
Temp_Min	confirmed
Rad_Max	unimportant
Rad_Min	confirmed
Rainfall_1	unimportant
Pressure_Max_2	confirmed
Pressure_Min_2	confirmed
Vapor_Max_2	confirmed
Vapor_Min_2	confirmed
Hum_Max_2	confirmed
Hum_Min_2	confirmed
Temp_Max_2	confirmed
Temp_Min_2	confirmed
Rad_Max_2	unimportant
Rad_Min_2	tentative
Rainfall_2	tentative
Pressure_Max_3	confirmed
Pressure_Min_3	confirmed
Vapor_Max_3	confirmed
Vapor_Min_3	confirmed
Hum_Max_3	confirmed
Hum_Min_3	confirmed
Temp_Max_3	confirmed
Temp_Min_3	confirmed
Rad_Max_3	confirmed
Rad_Min_3	confirmed
Rainfall_3	confirmed

After the feature selection the unimportant and tentative features were discarded and a new data set was created and linear regression also performed.

TABLE III. THE OUTPUT OF LINEAR REGRESSION AFTER THE FEATURE SELECTION

Coefficients	Estimate	Std. Error
Regression Constant	-0.00329	0.023221
Pressure_Max	0.134813	0.146964
Pressure_Min	0.059284	0.136995
Vapor_Max	-0.14409	0.114682
Vapor_Min	0.05452	0.111223
Hum_Max	0.051937	0.046166
Hum_Min	0.003889	0.070611
Temp_Max	-0.00853	0.066304
Temp_Min	0.135283	0.093188
Rad_Min	0.080893	0.03135
Pressure_Max_2	0.095757	0.158709
Pressure_Min_2	-0.16336	0.154165
Vapor_Max_2	0.12838	0.118011
Vapor_Min_2	0.292234	0.112391
Hum_Max_2	-0.11169	0.047715
Hum_Min_2	-0.15908	0.068584
Temp_Max_2	-0.00408	0.071758
Temp_Min_2	-0.11303	0.10533
Pressure_Max_3	-0.65068	0.148693
Pressure_Min_3	0.257571	0.157747
Vapor_Max_3	-0.16939	0.11424
Vapor_Min_3	-0.21864	0.110457
Hum_Max_3	0.18574	0.047743
Hum_Min_3	0.085519	0.068532
Temp_Max_3	-0.05347	0.070447
Temp_Min_3	0.078956	0.099122
Rad_Max_3	-0.1709	0.031131
Rad_Min_3	-0.00455	0.031815
Rainfall_3	0.032378	0.027234

This output of multiple linear regression also gives a linear e quation, which is mentioned below:

Rainfall = (Pressure_Max) * (0.134813) + Pressure_Min * (0.059284) + Vapor_Max * (-0.144087) + Vapor_Min * (0.05452) + Hum_Max * (0.051937) + Hum_Min * (0.003889) + Temp_Max * (-0.008531) + Temp_Min * (0.135283) + Rad_Min * (0.080893) + Pressure_Max_2 * (0.095757) + Pressure_Min_2 * (-0.16336) + Vapor_Max_2 * (0.12838) + Vapor_Min_2 * (0.292234) + Hum_Max_2 * (-0.111692) + Hum_Min_2 * (-0.159078) + Temp_Max_2 * (-0.004077) + Temp_Min_2 * (-0.113032) + Pressure_Max_3 * (-0.650682) + Pressure_Min_3 * (0.257571) + Vapor_Max_3 * (-0.169391) + Vapor_Min_3 * (-0.218637) + Hum_Max_3 * (0.18574) + Hum_Min_3 * (0.085519) + Temp_Max_3 * (-0.053472) + Temp_Min_3 * (0.078956) + Rad_Max_3 * (-0.170903) + Rad_Min_3 * (-0.00455) + Rainfall_3 * (0.032378) + (-0.003291) (3)

It was recorded that Residual standard error is 0.86 on 1405 degrees of freedom and Multiple R-squared value is 0.227, Adjusted R-squared value is 0.2115.The root mean squared error is 11.7406, which is also very high.

Then k-means clustering were performed on the above said data set. First the total within sum of squared calculated and wss plot was made by using elbow method. This method determined that the size of □k□ will be 4. The method was also validated by gap_stat method in R language to determine the number of cluster.

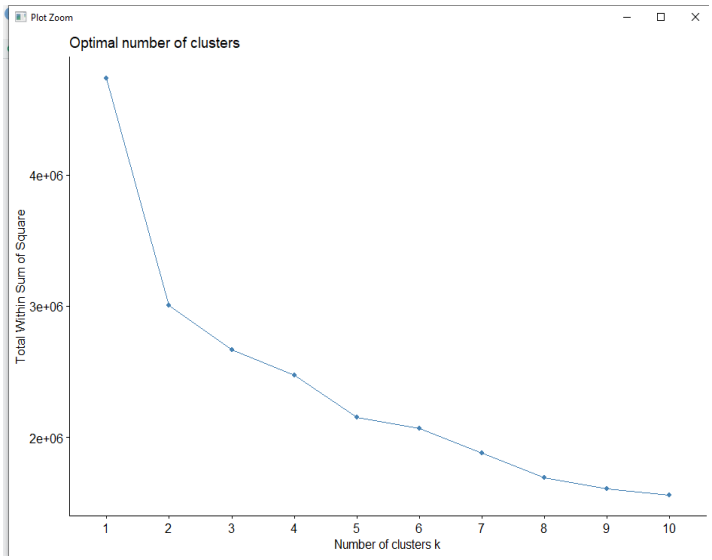


Fig. 2. Wss plot showing the optimum number of clusters

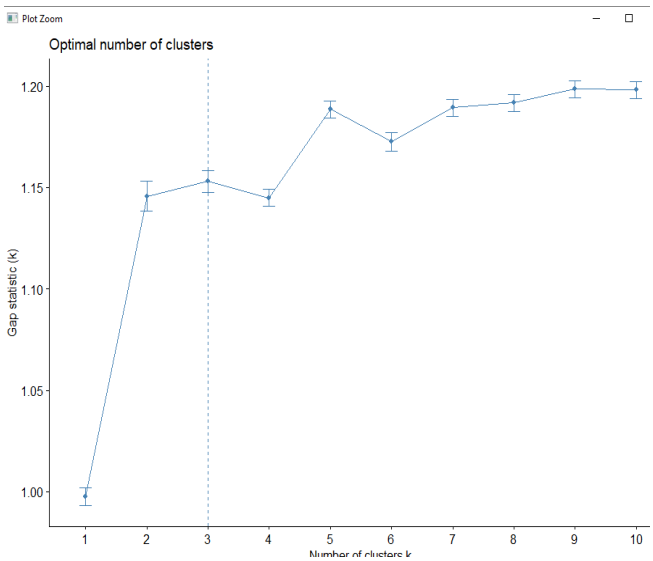


Fig. 3. Gap_stat method showing the optimum number of clusters

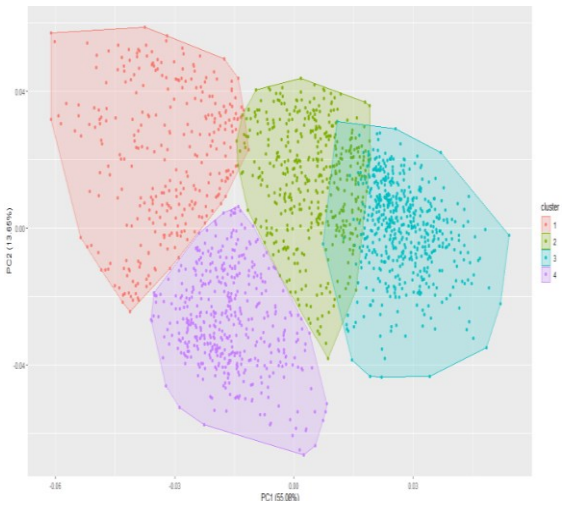


Fig. 4. Clusters of the data set (k-means where k=4)

Each cluster was fed into the model and multiple linear regression were applied again to achieve the result. A significant amount of accuracy were recorded in the output.

TABLE IV. THE OUTPUT OF LINEAR REGRESSION AT CLUSTER II

Coefficients	Estimate	Std. Error
Regression Constant	63.234355	159.459190
Pressure_Max	0.083326	0.304744
Pressure_Min	0.302526	0.288641
Vapor_Max	-0.05599	0.200713
Vapor_Min	-0.14914	0.185311
Hum_Max	-0.01152	0.051431
Hum_Min	0.057332	0.061827
Temp_Max	0.085435	0.233911
Temp_Min	0.449337	0.243745
Rad_Max	0.320937	0.698987
Rad_Min	0.281948	0.981226
Rainfall_1	0.012865	0.026139
Pressure_Max_2	0.07553	0.315234
Pressure_Min_2	-0.14694	0.32006
Vapor_Max_2	0.557232	0.199978
Vapor_Min_2	0.106149	0.185662
Hum_Max_2	-0.1728	0.051868
Hum_Min_2	-0.06442	0.06022
Temp_Max_2	0.068554	0.252792
Temp_Min_2	-0.35544	0.274286
Rad_Max_2	0.380665	0.720005
Rad_Min_2	0.913103	0.987394
Rainfall_2	-0.00706	0.026926
Pressure_Max_3	-0.97165	0.276571
Pressure_Min_3	-0.01137	0.30947
Vapor_Max_3	-0.5605	0.199342
Vapor_Min_3	-0.36748	0.186541
Hum_Max_3	0.266316	0.053075
Hum_Min_3	0.158933	0.060562
Temp_Max_3	-0.20494	0.253389
Temp_Min_3	0.258418	0.254571
Rad_Max_3	-2.84136	0.673489
Rad_Min_3	-2.04699	0.987444
Rainfall_3	0.058032	0.025154

The output of multiple linear regression gives a linear equation, which is mentioned below:

Rainfall = (Pressure_Max) * (-0.170394) + (Pressure_Min) * (0.261837) + (Vapor_Max) * (0.146748) + (Vapor_Min) * (-0.094494) + (Hum_Max) * (-0.052744) + (Hum_Min) * (0.001777) + (Temp_Max) * (-0.015656) + (Temp_Min) * (0.01272) + (Rad_Max) * (0.501028) + (Rad_Min) * (0.162517) + (Rainfall_1) * (0.04587) + (Pressure_Max_2) * (0.063313) + (Pressure_Min_2) * (0.039806) + (Vapor_Max_2) * (-0.034743) + (Vapor_Min_2) * (0.002312) + (Hum_Max_2) * (0.304489) + (Hum_Min_2) * (0.151876) + (Temp_Max_2) * (-0.540959) + (Temp_Min_2) * (0.16598) + (Rad_Max_2) * (-0.139332) + (Rad_Min_2) * (-0.185507) + (Rainfall_2) * (0.059424) + (Pressure_Max_3) * (0.122321) + (Pressure_Min_3) * (0.100949) + (Vapor_Max_3) * (-0.391693) + (Vapor_Min_3) * (1.257434) + (Hum_Max_3) * (2.082708) + (Hum_Min_3) * (-0.020157) + (Temp_Max_3) * (-0.20494) + (Temp_Min_3) * (0.258418) + (Rad_Max_3) * (-2.84136) + (Rad_Min_3) * (-2.04699) + (Rainfall_3) * (0.058032) +(63.234355) (4)

It was recorded that Residual standard error is 3.831 and Multiple R-squared value is 0.1505, Adjusted R-squared value is 0.04529. The root mean squared error is 2.320498, which is significantly lower than the previous experiment.

TABLE V. A COMPARISON BETWEEN THE OUTPUT BEFORE AND AFTER FEEDING THE MODEL

	Output of standard multiple linear regression	Output after the feature selection	Output after the data set was fed into the model
Residual standard error	11.54	0.86	3.831
Multiple R-squared	0. 2319	0.227	0.1505
Adjusted R-squared	0. 2138	0.2115	0. 04529
Root mean squared error	15. 83106	11.7406	2. 320498

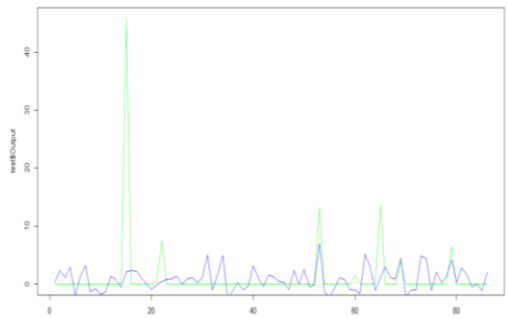


Fig. 5. Comparison between the actual and predicted data before fed into the model

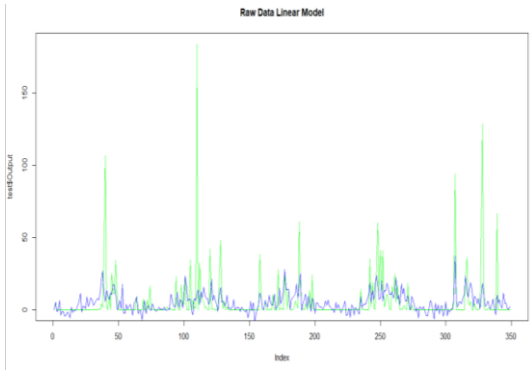


Fig. 6. Comparison between the actual and predicted data after fed into the model

V. CONCLUSIONS

This comparison talked about here has been created to foresee rainfall for a specific day dependent on the information of past three days. The meteorological information were gathered from Kolkata Meteorological center and utilized for investigation of the proposed model. A conglomeration of statistical and computational models were tried to figure the yield and this processed yield was contrasted and the objective yield for example rainfall. After testing these models, the following conclusions are made.

i. The clustering approach turns out to be an excellent tool that can predict the rainfall accurately by overcoming the seasonality effect on air pressure.
ii. The comparative model proposed here can be good alternatives for traditional meteorological approaches for weather forecasting. In the future works, the combined use of Feature selection and Artificial neural network may result in an excellent paradigm for prediction of air pressure. Moreover, we can also incorporate time series analysis in our data set to get more accurate result.
iii. In this paper the sensitivity analysis can also be incorporated to determine how different values of an independent variable affect a particular dependent variable under a given set of assumptions.

REFERENCES

Ahmed, H. A. Y., & Mohamed, S. W. (2021, February). Rainfall Prediction using Multiple Linear Regressions Model. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-5). IEEE.

Fayaz, S. A., Zaman, M., & Butt, M. A. (2022). Knowledge discovery in geographical sciences□a systematic survey of various machine learning algorithms for rainfall prediction. In *International Conference on Innovative Computing and Communications* (pp. 593-608). Springer, Singapore.

Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., & Akanbi, L. A. (2022). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, 100204.

Mohd, R., Butt, M. A., & Baba, M. Z. (2022). Grey wolf-based linear regression model for rainfall prediction. *International Journal of Information Technologies and Systems Approach (IJITSA)*, *15*(1), 1-18.

Duan, T., Anand, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A., & Schuler, A. (2020, November). Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning* (pp. 2690-2700). PMLR.

Roesch, I., & Günther, T. (2019, February). Visualization of neural network predictions for weather forecasting. In *Computer graphics forum* (Vol. 38, No. 1, pp. 209-220).
Das, R., Mishra, J., Mishra, S., & Pattnaik, P. K. (2022). Design of mathematical model for the prediction of rainfall. *Journal of Interdisciplinary Mathematics*, *25*(3), 587-613.

Gad, I., & Hosahalli, D. (2022). A comparative study of prediction and classification models on NCDC weather data. *International Journal of Computers and Applications*, *44*(5), 414-425.

A Comprehensive Study on Code Coverage Analysis for effective Test Development/Enhancement Methodology

Biswadeb Bandyopadhyay
Assistant Professor, Department of Computer Applications (CA)
University of Engineering and Management, Kolkata
biswadeb.bandyopadhyay@uem.edu.in

Abstract-The paper describes a tool developed by a team of testing practitioners dealing with a large collection of test suites used for testing Teradata which is large relational database system software. This team has studied the existing database software, available test suites, collected the code coverage data for a large number of available test suites for Teradata and developed an effective Web based Code coverage analysis tool to analyze these Code coverage performance of these test suite while running on Teradata to be able to understand code coverage of these test suites data in various possible dimensions. This activity was undertaken as part of a test engineering initiative to bring in place a set of innovative test engineering practices as potential business value adds.

1. Introduction

An effective way to measure the Quality of software product is the amount of code that has been tested (i.e. Code coverage). While this does not guarantee that the code is defect free, the risk of uncovering more defects from the customer's site is reduced considerably as more code is tested during the product test cycle. It should be realized that even 100% coverage does not guarantee a defect free code. Most Test engineer would agree that while one can never be sure of a bug free code, a significant milestone is achieved when "all the code has been tested." Code coverage can be a valuable measure, especially when time is taken to achieve a high coverage value.

The idea behind the code coverage is to improve the test cases by

- Identifying the uncovered code by the existing test suites, adding the new test cases and thereby improving the test suites.
- Identifying the redundant test cases in the existing test cases, thereby reducing the execution time of the test cases by removing them.

While working with these test suites, the team took an initiative to analyze the code coverage for all the available test suites to get some degree of confidence as to the existing level of code coverage. Code coverage provides a deep insight into the adequacy of the test cases and the need of or scope for improvement. The team undertook a comprehensive analysis of these test suites and collected code coverage data for a large number of such test suites. A Web based tool was developed where all these coverage data were stored, to order to do a comprehensive analysis of these code coverage data in various dimensions.

The Web based code coverage analysis tool developed by the team provided a convenient platform from where the user can obtain and analyze the code coverage data for various test suites. This proved to be an effective tool to quickly understand and analyze the test coverage scenarios.

The benefits of this tool were to be able to generate the following analysis reports

- Line level, function level and module level code coverage reports
- Annotated source code for function wise, module wise and test suite wise coverage data
- Annotated source code of a selected implementation file with lines hit, lines not hit, lines partially hit
- Analytical report of code coverage of a selected implementation file for various test suites
- To provide information on the most appropriate test suites to validate a bug fix/code

enhancements which will guarantee the maximum statement coverage of the file being added/modified

- To analyze any field reported problems, to identify whether the root cause of the failure was due to non-coverage of the code segment where the fix for the problem was found
- To identify the root cause of any regression problems due to any limitation of existing test suites used for regression testing

2. Why Rational's Pure Coverage:

There are lots of profiling tools in the market. But most of the tools do not support for server side, i.e. applications that will run continuously. They will generate coverage data only on graceful exit of the application. But Rational's Visual Purecoverage tool generates a detailed report of the code coverage, even the application terminates abnormally.

There two flavors of Purecoverage packages from Rational, one is on UNIX and another for Windows. On UNIX, it is necessary to instrument (adding Purecov option) at the time of compilation itself. So it would necessitate to re-build the package. But on NT we can instrument after the compilation, so there is no need to re-build the package. It is also possible to instrument any particular EXE/DLL that may be neede. But these should be built without any optimization options.

It is possible to save the coverage data of Purecoverage either in CFY format (only Purecoverage can open) or in ASCII Text format.

3. Implementation

Following section describes development of a code coverage analysis tool by a team of test practitioners who were involved in the development of test suites for testing a large relational database management system. The team utilized its experience of developing these test suites to extend it further to assess effectiveness of these test suites in terms of code coverage performance of these tests on the system under test. Essential idea was to develop such a code coverage analysis tool and utilize this tool in analyzing the test suites being developed, use that information for rationalizing the tests by eliminating the tests that are redundant, augmenting test suites with new test cases to improve the code coverage further, thereby enhancing the effectiveness of the test suites.

The test team developed a web based "Code Coverage Analysis tool" that provides a convenient platform from which the user can obtain and analyze the code coverage data for various Teradata Regression Test Suites. This data was collected using Rational's Purecoverage tool. The Code Coverage Analysis Tool provides all information such as percentage of DBS code coverage for different regression test suites, at module level, function level and line level. This provides complete information about overall code coverage performance of each individual test suites as a whole.

3.1 Minimum Cost – Maximum Coverage Model

To implement Minimum Cost - Maximum Coverage model for rationalizing Test suites and enhancing Code Coverage.

Cost parameters for optimization:

- Line Coverage (Ci)
- Number of Test Suites (Ti)
- Execution Time (Ri)

Min Cost –Max Coverage Model optimizes:
Min (Ti)=Max (Ci) & Min (Ri)

- Track the line coverage and minimize overlap across the tests
- Minimal test sets to cover Maximum source code

3.2 Tool is used by the Developers

- To identify the areas of code which are not covered or partially covered in order to improve the test cases
- To use the tool as a workbench for ensuring that the test cases they generate are adequate

3.3 Tool is used by the Testers

- To choose the Test Suites based on their percentage of coverage for a
- given source file
- To Correlate the Test Suites for their coverage for a given source file

3.4 Features of the Tool

Tool was built around Java Servlets, Java Server Pages, HTML, Apache Web Server, Jakarta-Tomcat, Carcase Version Control Software, Rational Pure Coverage Tool, Unix Shell scripts and Teradata RDBMS.

Feature includes:

- Data extraction
- Data loading
- Report generation with Code Coverage Analysis
- using HTML / Java Servlets / Apache Web Server / Jakarta-Tomcat / Teradata ODBC driver/ JDBC-ODBC bridge

Tool generates following reports:

- Correlating different Test Suites for Line-wise coverage for a given source file
- Function-wise coverage for different Test Suites for a given source file
- Pictorial presentation of Module-wise coverage for a given Test Suite

- Highlighting source file for coverage data with annotations

- Blue indicating covered lines
- Red indicating uncovered lines
- Pink indicating partially covered lines
- Showing list of Test Suites in ascending order of coverage for a given source file

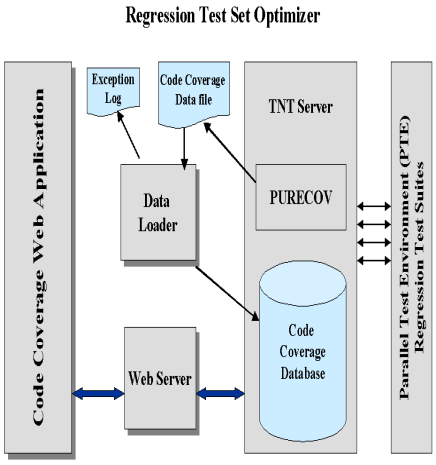


Figure 1 - Regression Test Set Optimizer

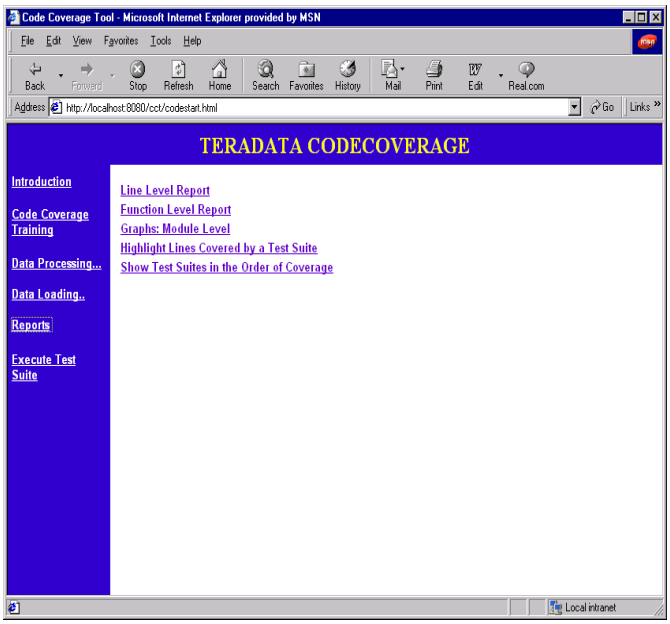


Figure 2 - Teradata Code Coverage Tool

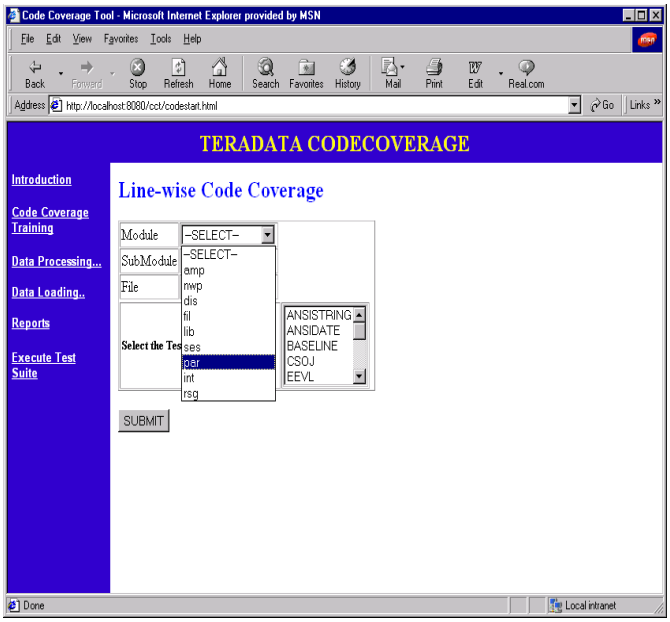


Figure 3 - Line Wise Code Coverage Menu

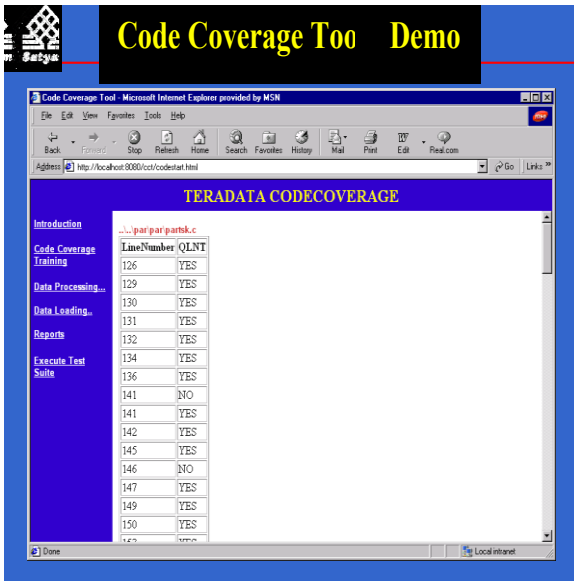


Figure 4 - Line Wise Code Coverage Output

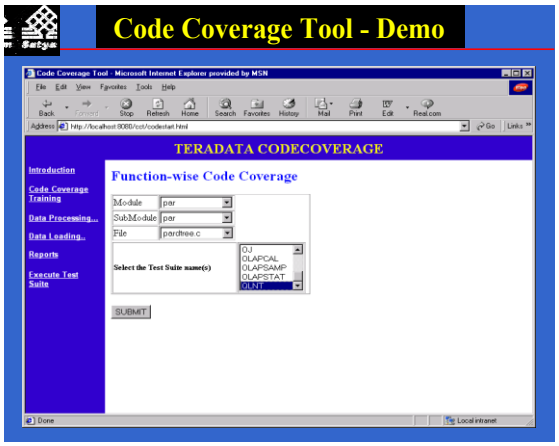


Figure 5 - Function Wise Code Coverage Menu

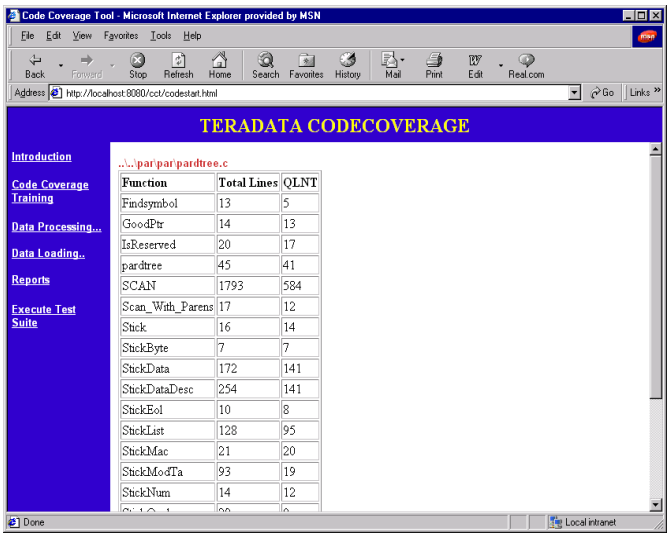


Figure 6 - Function Wise Code Coverage Output

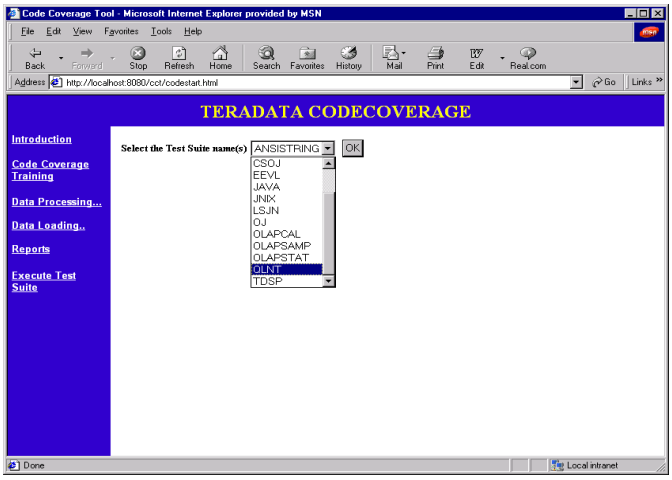


Figure 7 - Test Suite Wise Coverage Menu

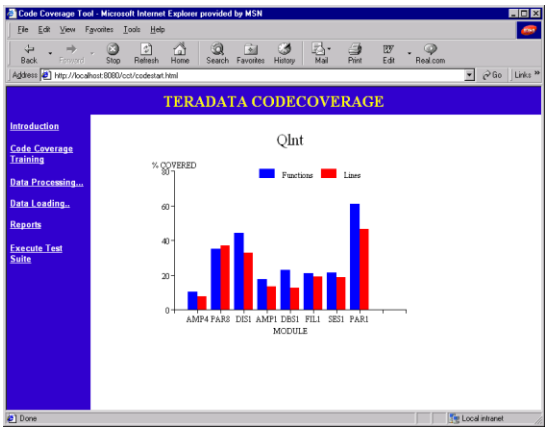


Figure 8 - Coverage Output of a particular Test Suite

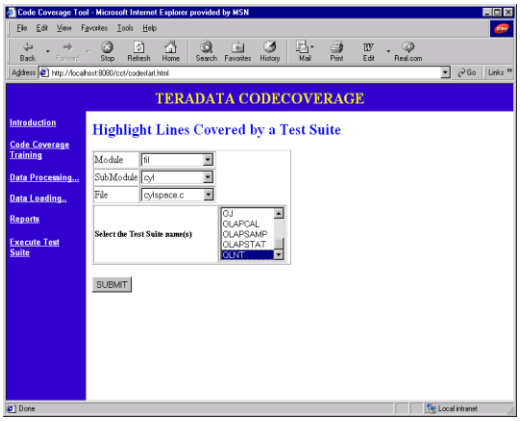


Figure 9 – Highlight Lines Covered by a Source File Menu

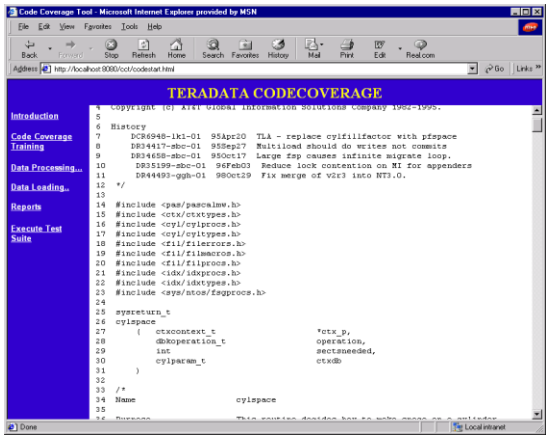


Figure 10 - Annotated Source Code of a File

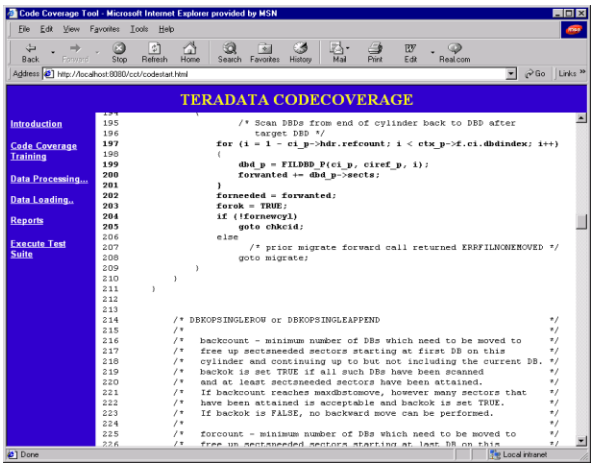


Figure 11 - Line Wise Coverage Output of a Source File

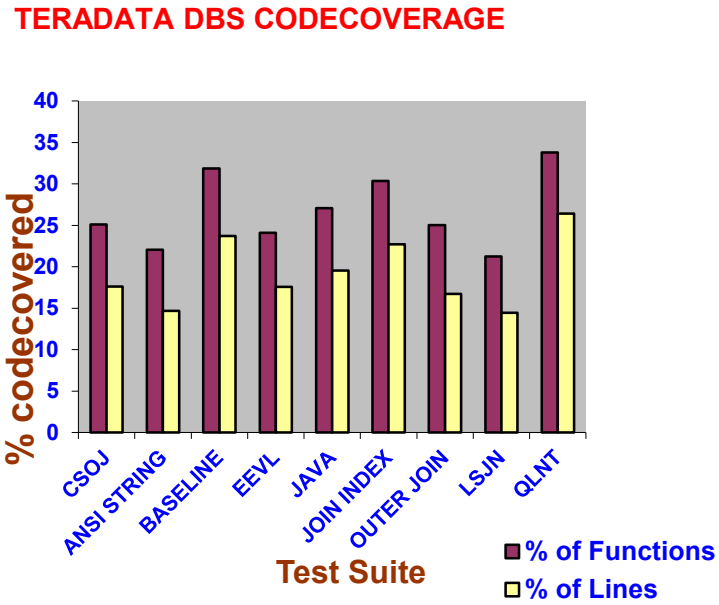


Figure 12 - Test Suite Wise Code Coverage Output

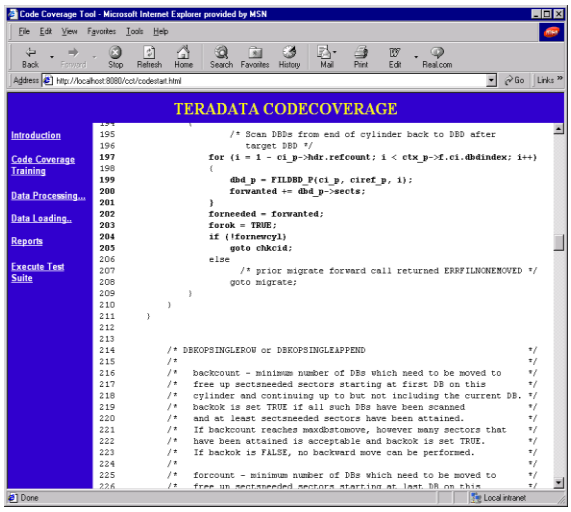


Figure 13 - Annotated Source Code of a File

4. Conclusion

Tool developed by the turned out to be quite effective in performing a number of activities. These included, Rationalizing the Test Suites, Generating Traceability Matrix, Improving Percentage of Code Coverage, Analysis of field encountered problems, to identify the root cause of any regression problems, due to any limitation of existing test suites used for regression testing, non-coverage of the code segment where the fix for a particular problem was found. The tool is unique in the sense that it focuses on optimizing the test suites from code coverage perspective of test suites, providing dual benefits simultaneously by improving code coverage and optimizing the test cases, which in turn results in reduction of test cycle time.

References

1. Effective Methods for Software Testing, William E. Perry, Willey Publication
2. Software Engineering: A Practitioner's Approach, Roger S. Pressman
3. Software Testing Concepts and Tools, Nageswara Rao Pusuluri
4. Software Testing in Real Worls, Edward Kit
5. Code Coverage vs Test Coverage: A Detailed Guide, Shreya Bose
6. Efficient use of code coverage in large-scale software development, Yong Woo Kim, CASCON '03: Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research
7. Code coverage analysis in practice for large systems, Yoram Adler; Noam Behar; Orna Raz; Onn Shehory; Nadav Steindler; Shmuel Ur; Aviad Zlotnick, 2011 33rd International Conference on Software Engineering (ICSE)
8. A REVIEW ON CODE COVERAGE ANALYSIS SARITA PATHY1 P.G. Dept. Of Computer Science and Application Jyoti Vihar, Sambalpur University, Burla, Sambalpur,Odisha,India