Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 44 (2023), No. 3, pp. 369–382
DOI : 10.47974/JIOS-1347

An ontological architecture for context data retrieval and ranking using SVM and DNN

Pooja Mudgil *
Department of Computer Science
Banasthali University
Rajasthan
India

Pooja Gupta †
Department of Computer Science and Technology
Maharaja Agrasen Institute of Technology
Delhi
India

Iti Mathur §
Nisheeth Joshi ‡
Department of Computer Science
Banasthali University
Rajasthan
India

Abstract

Context retrieval and ranking have always been an area of interest for researchers around the world. The ranking provides significance to the data that has to be presented in front of users but it also consumes time if the ranking architecture is not organized. The retrieval is dependent upon the co-relation among the data attributes that are supplied against a class label also referred to as ground truth and the ranking depends upon the sensing polarity that indicates the hold of the outcome towards asked information. This paper illustrates an ontological architecture that involves two phases namely context retrieval and ranking. The ranking phase is composed of three different algorithm architectures namely k-means, Support Vector Machines (SVM), and Deep Neural Networks (DNN). The DNN is tuned to fit and work as per the availability of a total number of samples. The proposed work has been evaluated for both quantitative and qualitative parameters in different sets and

* *E-mail:* **engineer.pooja90@gmail.com (Corresponding Author)**
† *E-mail:* **poojaguptamait@gmail.com**
§ *E-mail:* **mathur_iti@rediffmail.com**
‡ *E-mail:* **jnisheeth@banasthali.in**

scenarios. The proposed work has also been compared with other state of art techniques and is illustrated in the paper itself.

Subject Classification: *68T07.*

Keywords: *Context retrieval, Ranking, Ontology, Artificial intelligence, Trapdoor, Machine learning, Support vector machine, Deep neural network.*

1. Introduction

Information refers to a relevant substance in terms of data that can be utilized to conclude any substantial constraint. Due to the increasing interest of software firms like Facebook, and Google, content reliability got great attention, and the word was born referred to as context[1]. The ranking of contextual data also plays a significant role in data mining architecture. ML approaches have significantly marked their spot when it comes to Software Aided Designs (SADs). The SADs architecture is based on the training set, the rule engine, and the desired outcome viz. the context that is also referred to as the Ground Truth (GT) in this paper. The rule engine binds the input data attributes with rules that dignify the context or GT. To ensure reduced computation complexity, propagation-based learning algorithms have been observed often in practice[2]. It becomes necessary to rank the data once the context is known for the test set. Ontological architectures have been viewed as a solution to ranking problems in terms of computation complexity and relative polarity calculation of the retrieved content against a context.

An ontology is a shared and formal set of linguistic rule sets that can be applied to retrieve the co-relation among the context data that is stored in the repository and uploaded query set[3]. The ontology architecture develops a communication form that interacts between the context information and stored information[4]. To form ontology, there must be a definition of the ontology and the rule sets. Ranking refers to organizing the retrieved contextual data in such a manner that it integrates the dignity of the retrieved documents based on the potential that the data carry against the supplied query set. Though a lot of research in the field of retrieval and ranking has been published and is elaborated in the introduction and related work section as well, the contributions of the proposed workflow are listed as follows.

- A novel method of ranking inspired by propagational AI-based Neural Networks.

- A novel method of polarity generation using a Support Vector Machine.
- Evaluation of both quantitative and qualitative parameters for proposed work.

2. Related Work

Wongthongtham and Salih 2016 presented an ontology-based methodology for extracting textual data interpreted to define the data domain. To put it another way, the author conceptually analyzes sociological data in two stages: individual level and domain level. For idea verification, the authors selected Twitter as a reference media platform. In terms of accuracy and precision of context identification, the ontology-based approach makes use of mapping and ranking data and shows superior results[5]. Anil Kumar et. al. 2022 used Cellular Automata concept to secure and retrieve the textual content. The proposed model has been divided into two phases: incorporating the five machine learning classifiers for time-aware historical context analysis and the development of a framework classifier[6]. Sharma and Li 2019 proposed a unique self-supervised strategy for retrieving and extracting terms and keywords using end-to-end recurrent neural networks that were trained. Specifically, a text that self-identifies based on its context. Furthermore, the authors proposed using bidirectional context information and short-sentence corpus that were automatically labeled and used to construct the ground truth[7]. Zhu et. al. 2020, the authors presented an NN-based method that uses simultaneous long short-memory (LSTM) to describe the phrases. When compared to earlier systems, it effectively includes unsupervised learning and achieves superior results[8]. Kulmanov et. al. 2021, the authors define the ontologies to systematically define and explain the contextual data and can be used to handle the significant database. The authors provide an overview of the methods for ontologies to quantify similarity and integrate them into ML algorithms; in particular and calculate the similarity and ontology representation learning[3]. Bashir et. al. 2022 presented an emotion analysis architecture to predict emotions in Urdu. Ai et al 2018, the authors proposed a Deep Principal components-based Context Model using the underlying feature distributions, which will aid in fine-tuning the preliminary sorted list[9]. The algorithm architecture included the data collection process, annotation of the data along with the preprocessing and feature extraction. Tam et. al. 2021 also presented a Twitter sentiment-based analysis that combines Bi-LSTM with Convolution

Neural Network (CNN) for training and classification of twitter data[10]. Khalil et. al. 2021 presented emotion analysis for Arabic language using deep learning method architecture. The author used Bi-LSTM with forward and backward propagation model[11].

3. Proposed Work

The proposed work is derived in two phases for retrieval and ranking. The retrieval phase aims to extract the best possible context or GT value for the supplied user test data and the ranking phase arranges the retrieved data as per the polarity of the retrieved context. The work consists of users that can access the data from anywhere in the world and they are connected to a data center that manages the requests from the user. To process the context of the data, Machine Learning (ML) API is applied which uses ontological architecture to process the data when it comes to context retrieval. The inferences are listed in the X1 base which is the processing element. The evaluated context is analyzed in the stored generic database that contains system data files to be processed. The data files are extracted from the data repository and the second ontological architecture that uses the Levenberg principle to rank the data is applied. There is a neural network mechanism that propagates the data in the forward direction and updates the weights based on the Levenberg–Marquardt updating principle[12].

The working principle Levenberg consists of the first layers taking the data elements as inputs and generating initial propagation weight for any further propagation. The middle layer, viz. te hidden layer, uses an activation function in which each element $x_i \in \{x_1, x_2 \ldots . x_n\}$ is assigned to a new weight value w_i based on the previous weight value initiated at the input layer. The hidden layer also consists of a transformation function that maps the data x_i with its updated weight w_i against its class category. To validate the training, Levenberg uses gradient and Mean Squared Error (MSE).The gradient is generated by splitting the data into training, validation, and test data. When the least MSE against each class category is attained, the gradient is expected to be satisfied. The system validates the MSE with each propagation which is also termed an epoch in the Levenberg architecture method. The stopping criteria of the Levenberg model are based on either the success of the gradient or the end of total supplied epochs. The last layer produces the filtered trained architecture against the ground truth values. The trained data is used to classify the test data. Conjugate Based Neural Network(CBNN) follows the same principle

though it adds entropy along with MSE in the stopping criteria[13,14]. For a small set of samples, CBNN performs more efficiently as compared to Levenberg architecture and hence CBNN has been used in the ranking phase rather than in the context retrieval phase. Finally, the ranked data is supplied to the user via the data center.

3.1 *The Context Retrieval*

The ontological architecture uses similarity indexes namely cosine similarity and Euclidean distance in order to pass it to the learning mechanism for context verification as shown in Figure 1. Along with similarity indexes, Term Frequency (Tf) and Inverse Document Frequency (Idf) have also been evaluated. The data is first passed to the pre-processing where the Porter Stemming Algorithm removes the words that do not fit the list.

In order to train the system, Ontology 1 uses Feed Forward Back Propagation Neural Network which intakes 70% of the total data for the training process where as the rest of the 30% is considered to be the test data. To pass the data for the training, five features Tf, Idf, Cosine Similarity, Euclidian Distance, and Dice Coefficient have been evaluated. To select the best suitable data Principle Component Analysis (PCA) has

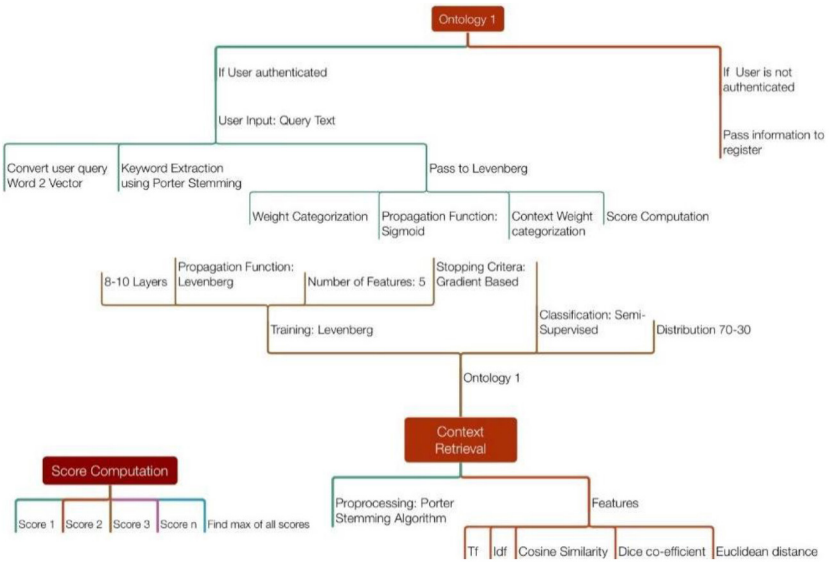

Figure 1

The ontological architecture of context processing

been implemented. The algorithm architecture for the training portion is illustrated using Context Training and Retrieval Algorithm.

The FeedForward Back Propagation NN has been initiated with a single neural layer and increased during processing. If the value of the gradient of feedforward is less than (E_G) expected gradient used for training purposes, then the stored value is the function of fn. The proposed algorithm has been applied using MATLAB's NN toolbox and hence E_G and Attained Gradient (A_G) both are calculated using the NN toolbox itself. PCA results in 70% sample selection as the final result. The algorithm also intakes the user query as 'Uq' and applies the Porter Stemming Algorithm and transforms the data vector into a feature vector similar to that of training data. The query context is simulated against the supplied number of trained contexts and based on the neural classified context, the context to be ranked is evaluated. The ranking process of the data is illustrated in the next section.

Algorithm 1 Context Training and Retrieval

Require: fv, k

Output: RC where RC is the retrieved category, fv is feature vector, k is total number of classes

Ensure: $y = Optimal(fv.gt)$

For $i = 1$ to k

$$fv_c \leftarrow fvk$$

$$O_f v_c \leftarrow ApplyPCA(fv_c)$$

$Store(O_f v_c) fv_c$ //features of category

Initiate Levenberg Feed forward back Propagation

Require: l is total layer

E_G : $Get(Gradient)(Sigmoid(fv))$ //get the expected gradient from the sigmoid function of Levenberg

$ep_r = 0$ //current running epochs epoch = 100, Maximum epochs length

$$While\ A_G \le E_G\ AND\ ep_r \le (epochs)\ do$$

$$ft = fx(gt.Update(Weight).Sigmoid, l)$$

A_G is the evaluated gradient based on the current weight

$$A_G = ft$$

A_T is the gradient to feedforward

ep_r ++

$$End_{while}$$

$$End_{for}$$

U_q : Take Query from User

t_v: transformed vector

$$RC : ContextSimulate(U_q, ft)$$

Return RC

3.2 *Ontological Ranking of context data*

Once the original context has been retrieved from the repository, the proposed algorithm aims to rank the data based on the applied hybrid mechanics of proposed algorithm that utilizes a combination of Support Vector Machines (SVM) and Conjugate Based Neural Network (CBNN). When the context is retrieved, the proposed algorithm creates a hybrid index in the sequence of ontological architecture. The ranking ontology is divided into two on to logical processes namely Ranking Ontology Process R1(ROPR1) and Ranking Ontology Process R2(ROPR2). ROPR1 uses Support Vector Machine (SVM) in order to generate the hyperplane that suits the retrieved context for ranking. select the hyperplane of SVM, the proposed algorithm utilizes k-means and statistical machine learning parameters that are illustrated in ROPR1-Algorithm.

Algorithm 2 ROPR1

Require: fv_c

Output: s_v //where s_v is the support vectors

$k = 2$//where k is total number of groups to be considered

$R - MSE_A$ = []//initialize all R-MSE to empty, divide the data in two groups

$$[id, cent] = kmeans(fv_c, 2)$$

$$for\, i = 1 to\, k$$

$f = find[id == i]$ //find all the features that belongs to group1

$$c = cent[i]$$

$$R - MSE_A = R - MSE(c, f)$$

$$d_p = (R - MSE_{A[1]} - R - MSE_A[2] / R - MSE_A[1] * 100$$

$$if\ d_p > 50$$

$k_t = 1$ //apply linear kernel where k_t *is kernel choice*

Else

$k_t = 2$ //apply polynomial kernel where k_t is kernel choice

$$End_{if}$$

$$SVM_s = initiate(SVM(fv_c, k_t))$$

$$End_{for}$$

$$s_v : ExtractSupportVectors(SVM_s)$$

Return s_v

ROPR1 divides the entire feature vector into 2 clusters using k-means. As the retrieved context is now fixed, the data has to be ranked only. In order to rank the context data, proposed algorithm uses a hybrid structure of SVM and Deep Neural Network(DNN)[16,17]. As SVM is a binary classifier and the data can only be further separated into two classes. The cluster group is stored into variable 'id' and the centroids of the respective clusters have been stored into variable 'cent'. ROPR1 calculates the Root Mean Squared Error (RMSE) of the cluster elements and its respective centroid for each class and stores it to $RMSE_A$. If the difference between the calculated RMSEs is more than 50%, it denotes that the data in each group is distinct and can be separated using linear kernel, else polynomial kernel will be utilized. ROPR1 finally initiates the SVM with the selected kernel k_t by the ROPR1 and extracts the support vectors as s_v only over the provided data. 50% separation demonstrate a complete separated class architecture and hence it is used as margin threshold. It is flexible in nature but for the proposed work value, it works well with 50% margin. These support vectors will be used to further train CBNN for final ranking[18]. The algorithmic architecture for CBNN is presented in Algorithm ROPR2.

Algorithm 3 ROPR2

Require: s_v where s_v is the support vector attained by ROPR1

Output: Classified Ranking Scores as scores

$LC = \log(a + \sqrt{dr/dc})$ where LC is layer count dr and dc are total number of data rows in every class and dc is total number of attributes in the list

$s_f = 100$ //take sampling frequency to 100

$$sf = 1$$

$scores = []$ //initiate an empty score for ranking

$While(sf < s_f)$ do

$pop = Random(pop)(s_v, 70/100)$ //select a random population integrating 70% of total population

$$Borrow[id, cent] \; from \; ROPR1$$

$id_p = id.pop$ //Extract the ids of the population in id_p

Initialize $CBNN_s = DNN(pop, id_p, LC)$ //initialize DNN with selected population, their respective groups borrowed from ROPR1 and total number of layers

$rc = 1$ //record number

$While(rc < id_p.count)$ do

$$C_1 = Classify(CBNN_s, s_v[id_p[rc]]$$

$$score[sf, rc] = C_1.score \; // \text{extract the classification score}$$

$$rc++$$

$$End_{while}$$

$$sf++$$

$$End_{while}$$

$$Return \; scores$$

ROPR1 takes the advantage of the support vectors to compute the layer count that is to be utilized in CBNN as LC. LC utilizes the total number of identities or data rows that have been extracted by the SVM as dr and the total number of attributes as dc. ROPR2 takes a total sampling frequency of 100 in which with each sampling value, a random population that is 70% of the total rows that have been selected by SVM will be generated. The identities of the support vectors have been borrowed from ROPR1 as id. The ROPR1 trains CBNN with sampled identities and the generated population along with the total number of supplied layers. The trained architecture is stored in $CBNN_s$ that contains the hybrid index of the stored data generated by the combination of SVM and CBNN, and

each record will that has been considered by ROPR1 will be classified against $CBNN_s$ for 100 samplings. Thus, ROPR1 and ROPR2 coordinate to produce the best possible score and classification or the ranking outcomes. More details of the steps followed in the two algorithms are summarized in Algorithm ROPR1 and Algorithm ROPR2. For every classified set, the classification score is generated and stored in a variable named scores. The resultant scores will be ordered in descending order such that the highest score record set will be ranked 1 and the lowest score value record will be kept at the end of the returned set.

4. Results and Discussion

The proposed work model has both quantitative and qualitative parameters for analysis. The quantitative parameters viz. Precision, Recall, Accuracy and F-Measure shown in Table 1 has been done using three datasets.

- Sentiment 140 datasets retrieved from https://www.kaggle.com/datasets/kazanova/sentiment140.
- SemEval 2013 dataset retrieved from https://www.kaggle.com/datasets/azzouza2018/semevaldatadets.
- Covid19 Tweets dataset retrieved from https://github.com/mykabir/ COVID19.

Table 1

Average Performance Score

Average Scores	Proposed	Bashir et. al.[15]	Tam et. al.[10]	Khalil et. al.[11]
Precision	.86135896	.79142726	.77905418	.79112518
Recall	.846	.7713	.7594	.7717
F-measure	.85382	.77989	.76817	.77935
Accuracy	89.989	82.05126	82.40124	82.687

The average precision is .861358 which is 8.83% more efficient than Bashir where 10.56% more efficient than Tam and 8.87% more efficient than Khalil. In a similar fashion average proposed recall is .846 which is 9.68% more efficient than Bashir whereas 11.40% more efficient than Tam and 9.62% more efficient than Khalil. When it comes to qualitative parameters, they refer to the architecture of ranking the results. The evaluation is illustrated as follows.

- Ranking time: It is the computation time to rank the outcome values.
- Trapdoor time: It is the computation time for one complete process.

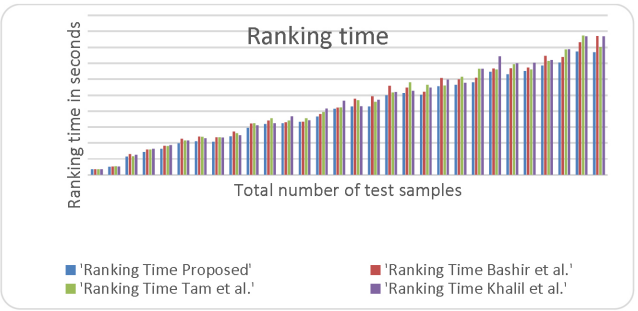

Figure 2

Ranking time

The trapdoor time involves the ranking time and the classification time. The classification time is marginal due to the efficient classification architecture of the proposed algorithm. The ranking time increases with the raise in the total number of test documents as it would take more time to rank more data.

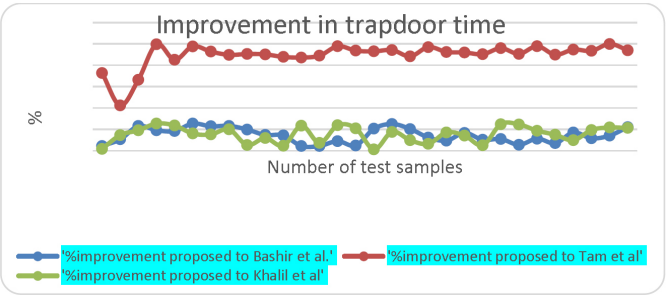

Figure 3

% Improvement

For a maximum of 4500 test samples, the ranking time for the proposed algorithm is 15.38 seconds whereas Bashir et. al. consumed 17.40 seconds. Tam et. al. demonstrated 16.01 seconds in total whereas Khalil et. al. ended up at 17.36 shown in Figure 2.The maximum trapdoor time for the proposed case scenario is 16.70 seconds for 4500 test samples whereas Tam et. al. consumed the maximum time i.e 31.49 seconds. The improvement in trapdoor time of the proposed algorithm to other state of art techniques is shown in Figure 3. The proposed algorithm outcast the existing state of art techniques. When it comes to Bashir et. al. and Khalil

et. al., the improvement is $\approx 9\%$ whereas in comparison to Tam et. al., the proposed work is $\approx 33\%$ more improved.

5. Conclusion

The paper has presented an improved ranking of the test samples in light of ontology-based architecture. The work encompasses three different datasets for the evaluation of the proposed work and a mixed query set of 4500 samples have been used to test and rank the trained samples. The proposed work has been evaluated for precision, recall, and f-measure and outcast state of art techniques by $\approx 8\%$ in terms of precision, recall, and f-measure whereas a maximum of 33% improvement has been recorded when it comes to computation complexity. The proposed work outcast the other state of art techniques by significant margins. Trapdoor time reflects the overall time involving both ranking and classification time. Thus, to evaluate the performance of the ranking architecture, the proposed algorithm has been evaluated for a ranking time as well as the trapdoor time. Overall, the proposed architecture exhibited an improvement that varies between $\approx 9\%$ and $\approx 33\%$ evaluated over three existing studies.

References

[1] A. Muhammad, N. Wiratunga and R. Lothian, Contextual sentiment analysis for social media genres, Knowledge-based systems, 108, pp. 92-101 (2016).

[2] J.S. Garrido, D. Dold and J. Frank, Machine learning on knowledge graphs for context-aware security monitoring, in 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 55-60 (2021).

[3] M. Kulmanov, F.Z. Smaili, X. Gao and R. Hoehndorf, Semantic similarity and machine learning with ontologies, Briefings in Bioinformatics, 22, pp. 1-18 (2021).

[4] S. Malik and S. Jain, Ontology based context aware model, ICCIDS 2017 - International Conference on Computational Intelligence in Data Science, Proceedings, 2018-January, pp. 1-6 (2018).

[5] P. Wongthongtham and B.A. Salih, Ontology-based approach for identifying the credibility domain in social Big Data, *Journal of Organizational Computing and Electronic Commerce*, 28, pp. 354-377 (2018).

[6] Anil Kumar & Sandeep Kumar Sharma. Information cryptography using cellular automata and digital image processing, *Journal of Discrete Mathematical Sciences and Cryptography*, 25:4, 1105-1111 (2022), DOI: 10.1080/09720529.2022.2072437.

[7] P. Sharma and Y. Li, Self-supervised contextual keyword and keyphrase retrieval with self-labelling (2019).

[8] X. Zhu, C. Lyu, D. Ji, H. Liao and F. Li, Deep neural model with self-training for scientific keyphrase extraction, *PLOS ONE*, 15(5): e0232547 (2020).

[9] Q. Ai, K. Bi, Um. Amherst Amherst, J. Guo, W. Bruce Croft CICS, Q. Ai et. al., Learning a Deep Listwise Context Model for Ranking Refinement, The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 18, pp. 135-144 (2018).

[10] S. Tam, R. Ben Said and Ö.Ö. Tanriöver, A ConvBiLSTM deep learning model-based approach for Twitter sentiment classification, IEEE Access, 9, pp. 41283-41293 (2021).

[11] E.A.H. Khalil, E.M.F. El Houby and H.K. Mohamed, Deep Learning for emotion analysis in Arabic Tweeets, *Journal of Big data*, 8(1), pp. 1-15 (2021).

[12] R. Zhao and J. Fan, On a new Updating Rule of the Levenberg-Marquadrt Parameter, *Journal of Scientific Computing*, 74, pp. 1146-1162 (2018).

[13] B. Zhang, Y. Liu, J. Cao, S. Wu and J. Wang, Fully complex conjugate gradient-based neural networks using Wirtinger calculus framework: Deterministic convergence and its application, Neural Networks, 115, pp. 50-64 (2019).

[14] A.G. Farizawani, M. Puteh, Y. Marina and A. Rivaie, A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches, in *Journal of Physics: Conference Series*, 1529, pp. 22040 (2020).

[15] M.F. Bashir, A.R. Javed, M.U. Arshad, T.R. Gadekallu, W. Shahzad and M.O. Beg, Context aware emotion detection from low resource urdu language using deep neural network, Transactions on Asian and Low-Resource Language Information Processing (2022).

[16] A. Upadhyay, S. Nadar and R. Jadhav, Comparative study of SVM and KNN for signature verification, *Journal of Statistics and Management Systems*, 23(2), pp. 191-198 (2020).

[17] H. Patel, A. Thakkar, M. Pandya and K. Makwana, Neural Network with deep learning architectures, *Journal of Information and Optimization Sciences*, 39(1), pp. 31-38 (2018).

[18] H. Iidukaand Y. Kobayashi, Training deep Neural Networks using Conjugate gradient-like methods, Electronics, 9(11), pp. 1809 (2020).