Improving Obsolescence Detection Accuracy using Recurrent Neural Networks

Manasvi Gurnaney, Shubhangi Neware

Abstract*: Forecasting a product's obsolescence depends on a multitude of factors which can be both technical and non-technical aspects of the product under study. The predictions are usually an approximate of the obsolescence and might not reflect the true nature of the product. Thus, researchers from various fields including market research, technology, public perception and others unite together in order to device a model which can be used for efficient obsolescence detection of products. In this paper, we propose an algorithm for effective obsolescence detection with the help of integrated datasets and a recurrent neural network (RNN). The RNN is used so that the effectiveness of prediction can be improved, and it is found that RNN is better when compared with other standard prediction classifiers.

***Index Terms*: Obsolescence, recurrent neural network, perception, prediction**

I. INTRODUCTION

Accurately predicting that how long a product will be in the market is a boon for manufacturers and product designers. Based on these predictions, the product designers can arrange for better improvements in the product, so that the lifetime of the product can be improved in the running market. In order to perform this prediction (also called as obsolescence detection), the researchers usually follow the given steps,

A. Collection of similar product data from multiple sources

Data collection is one of the most crucial aspects in prediction of product lifetime. Collecting data from multiple sources is a must, so that the complete information about the product can be obtained, and a comprehensive product study model can be developed. For example, in the mobile phone domain, it is essential not only to collect the technical specs of the mobile, but also to collect the market reviews, the customer opinions on how the phone is performing, etc. All these details assist the prediction engine to obtain a better result in terms of prediction accuracy for the lifetime of the smartphone. Researchers can collect this data from standard company datasets like Samsung, Apple, etc., and market research websites such as gadgetsnow.com, ndtv.com, GSMArena.com, fonearena.com etc. in order to generate a proper dataset for prediction.

B. Pre-processing of the data in order to remove any outliers

The data obtained from these sources is usually in a very non-standard format, and needs a lot of pre-processing in order to make it research usable. Thus, several pre-processing techniques like outlier removal, clustering, missing value restoration and others are applied so that the input dataset can be cleaned and a proper resultant dataset can be obtained. The output dataset after pre-processing can have both numerical and non-numerical values. For example, if the dataset collected has the number of bands the smartphone supports, then these bands can be 3G/LTE/4G among others, thus this data can be in non-numeric form, while other parameters like memory size, processor speed and battery capacity will be in numeric format.

C. Feature extraction from the dataset

The pre-processed dataset is given for feature extraction. In this process, the features which have similar properties are clubbed together, while the features which do not change much w.r.t. the entries in the dataset are removed. For example, the features like number of bands supported and number of WiFi modes supported are usually dependent on one another, thus they are combined to form a single feature set, while the features like screen size do not change much with each product, thus it can be discarded in order to have sufficient variation in the feature values w.r.t. each individual product. This processed feature dataset must contain the values which might be related directly or indirectly to the product obsolescence value, so that the system can find a relation between these features and lifetime of the product.

D. Conversion of features into numerical values

Generally all features are not in numerical format, as stated previously. Thus, there is a need to convert the non-numerical entities into numerical format so that the prediction model can be trained correctly. In order to do that task, we need to use a customized text to numeric value converter, which can represent each unique text value with it's numeric counterpart. For example, if we want to convert the supported network bands from the input dataset into numerical values,

Revised Manuscript Received on May 24, 2019.

Manasvi Gurnaney, Computer Science and Engineering Dept, Shri Ramdeobaba College of Engineering and Management, Nagpur, India.

Dr. Shubhangi Neware, Computer Science and Engineering Dept Shri Ramdeobaba College of Engineering and Management, Nagpur, India.

then it can be done via the following table,

Table I. Convert input dataset to numerical values

3G	4G	LTE	GPRS	5G
1	2	3	4	5

Thus, all the smartphones which support 3G will be given a value of 1 under the supported network column, and so on. This makes sure that the complete dataset is numerical and allows the prediction engine to be trained properly.

E. Training the prediction engine

Training the prediction engine properly via correct selection of training parameters is very crucial. Algorithms like neural networks require careful selection of the number of layers, the number of neurons, activation functions, training and testing set division, etc. While algorithms like particle swarm optimization, need proper selection of number of particles, stopping criteria, fitness function, etc. Once these parameters are selected, then the engine is trained and kept ready for prediction.

F. Predicting the obsolescence of the product from the trained data

The trained engine is then used for prediction of product lifetime based on the number of parameters to be used for prediction, the values of these parameters and the actual v/s predicted life time. If the prediction accuracy is not above a set threshold, then the layer is again trained with the help of the previous step, and re-evaluation of accuracy is done in order to get a sufficiently high accuracy trained engine.

In the next section, we describe various methods for obsolescence detection, followed by the proposed recurrent neural network based architecture for lifetime prediction, and conclude this text with some interesting observations about the developed technique and some future research works which can be carried out by the researchers.

II. LITERATURE REVIEW

Determining out of date quality is receptive in nature and depended on the goals of the issue once took note. The most traditional methodologies incorporate lifetime or last-time purchase (Rojo et al., 2010). There are two sorts of estimating strategies, specifically gauging of the out of date quality hazard and anticipating of the outdated nature date (life cycle determining). Outdated nature hazard determining is utilized to anticipate the likelihood that a part still underway (Josias, Terpenny, and McLean, 2004; Rojo et al., 2012). A couple of scientists centre around the forecast of the danger of out of date quality. In this unique situation, Rojo et al. (2012) directed a Delphi concentrate to break down the danger of outdated nature. They built up a hazard utilizing a few pointers, which are; years to finish of life, the quantity of sources accessible, and the utilization rate versus accessibility of the stock. Another methodology created by Josias et al. (2004) plans to make a hazard record by estimating the quantity of sources, life cycle arrange (presentation, development, development, decay, end of life), and market chance. (van Jaarsveld and Dekker, 2011) built up a technique dependent on authentic interest information to gauge the danger of out of date quality. The danger of outdated nature was assessed dependent on Markov Chain. Last, (Grichi et al., 2017; Jennings et al., 2016) have utilized information driven strategy by make AI calculations to estimate the outdated nature danger of an expansive number of parts. On the other hand, forever cycle gauging, Solomon et al. (2000) were the first to present the existence cycle estimating strategy. In their paper, the scientists led an investigation to foresee the existence phase of a section from the existence cycle bend, which included six phases: presentation, development, development, immersion, decrease, and out of date quality. Another strategy was created by (P. Sandborn, 2007) utilizing information mining with Gaussian strategy to anticipate the zone of oldness. This zone is given somewhere in the range of +2.5σ and +3.5σ and gives time interims for the period for the part will end up old. In addition, different specialists have acquainted relapse investigation with foresee the date of outdated nature (Gao et al., 2011).

Irregular backwoods are a joining of tree indicators where each tree relies upon the estimations of an arbitrary vector independently (Breiman, 2001). A comparable dispersion applies for every one of the trees in the backwoods. The tree classifier of a woods has a speculation blunder which depends on the solid connection between's everything trees in the timberland. Arrangement precision increments essentially when the gathering of trees is expanded. An essential model is packing, where to raise each tree, a self-assertive choice (without substitution) is done from the set precedents. Another precedent is irregular part choice where subjectively, the split is chosen from among the K best parts at each and every hub. An arbitrary woodland calculation comprises of pivoting numerous choice trees that are haphazardly developed and after that producing them. Bootstrap testing (OOB: Out-Of-Bag inspecting) is utilized in RF to have a superior gauge of the dissemination of the first informational index. Surely, bootstrapping implies haphazardly choosing a subset of the information for each tree as opposed to utilizing every one of the information to assemble the trees. In measurable terms, if the trees are uncorrelated, this lessens the conjecture fluctuation. The principle favorable position of arbitrary backwoods is their protection from changes and predispositions. The arbitrary woodland calculation is utilized in the relapse case to foresee a nonstop reliance and arrangement variable so as to anticipate an absolute ward variable. For the relapse type, an arbitrary backwoods comprises of a lot of straightforward forecast trees; each is fit for creating a numerical reaction when given a subset of informative factors or indicators. The blunder in this conjecture is gotten Out Of Bag (OOB). PSO called swarm insight or aggregate knowledge is created by Eberhart and Kennedy in 1995 (Kennedy and Eberhart, 1995; Shi, 2001). The general conduct of the

PSO isn't customized ahead of time however rises up out of the grouping of rudimentary associations between people.

In this unique situation, numerous scientists have connected the PSO in a few AI for enhancement (Lin, Ying, Chen, and Lee, 2008; Xiaodan, 2017). The Optimization strategy for PSO can be iterative; every molecule comprises of changing the speed toward its wellness esteem and worldwide adaptation of PSO. For the development, the molecule must settle on its next development (its new speed) by straightly consolidating three snippets of data: its present speed $Vij\ n$ (speed), its best execution effectively discovered $Pij\ n$ and which is known as the individual best position (pbest), and the best execution of its neighbors or sources $Pgj\ n$ known as the worldwide best position (gbest). The following area portrays the proposed methodology for obsolescence location dependent on recurrent neural networks.

III. PROPOSED RECURRENT NEURAL NETWORK BASED PREDICTION ENGINE

In this section, we describe the proposed recurrent neural network based architecture for prediction of product lifetime. The architecture can be described using the following diagram,

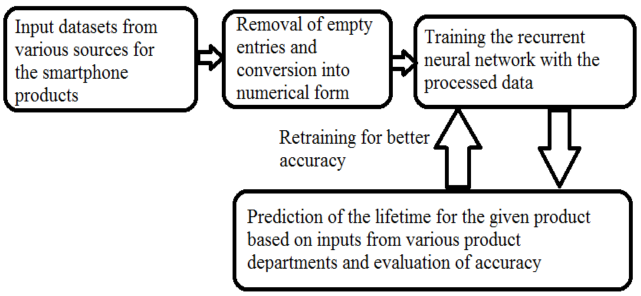

Figure 1. Overall System Architecture

The input datasets are collected from gsmarena.com website and from the fonearena.com websites manually, by visiting each product page for smartphones and maintaining that data in a comma separated value format. This data is very unstructured, some products have some unique fields while some other field are common. For example, the product NGage QD from Nokia, had a very different structure of device, and has hot-swappable option for the SD Card, while other products do not have this field. In order to tackle this, we kept only the common parameters from each of the smartphones, and obtained nearly 40 fields for about 2500 different smartphones. These fields are given as follows:

Table II. Features of Dataset

brand	status	Chipset	bluetooth
model	dimentions	GPU	GPS
network_technology	weight_g	memory_card	NFC
2G_bands	weight_oz	internal_memory	radio
3G_bands	SIM	RAM	USB
4G_bands	display_type	primary_camera	sensors
network_speed	display_resolution	secondary_camera	battery
GPRS	display_size	loud_speaker	colors
EDGE	OS	audio_jack	approx_price_EUR
announced	CPU	WLAN	img_url

The common parameters from these 40 fields are merged together, and parameters like img_url and NFC are removed from the dataset, as they are not useful for any prediction analysis. The parameters network_technology, 2G_bands, 3G_bands, 4G_bands and network_speed are combined together, display_type, 4display_resolution & display_size are combined together, CPU, Chipset & GPU are combined together, memory_card and internal_memory are combined, and finally rimary_camera & secondary_camera are combined in order to obtain a total of 22 unique features for the smartphone dataset.

These features are then converted from text format to numerical format in order to get the final dataset of numerical values for all the smartphones. Once the dataset is prepared we appended it with the lifetime field, which was manually researched for each of the smartphone from various online and company sources. After performing all these steps, the final dataset was prepared for prediction purposes.

Recurrent neural networks (RNNs) are dynamical frameworks that make productive utilization of fleeting data in the info succession, both for characterization just as for expectation. This implies in the wake of preparing, connections between the present info and inward states are handled to create the yield and to speak to the significant past data in the inside states. Amid a managed learning method the objective qualities comprise a second wellspring of data. These objective qualities indicate the important connections in the information grouping. The inward elements and info arrangement of a recurrent network decide the neurons' actuations. In this way, after reasonable preparing of the loads, the initiations of yielded neurons are utilized to describe and arrange an introduced info succession. To improve the arrangement capacity as far as learning velocity and speculation, we incorporate a period arrangement expectation undertaking amid preparing. This extra assignment extricates increasingly agent highlights from the given information. Along these lines the assistant errand fills in as an extra wellspring of data for the learning procedure. The accompanying figure shows the structure of the RNN being utilized for grouping,

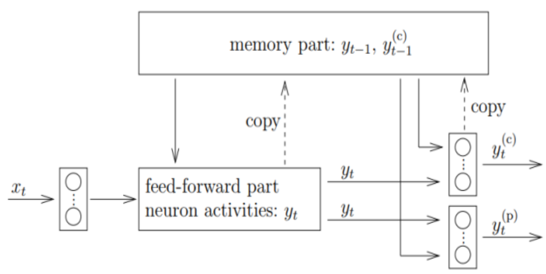

Figure 2. RNN Network Structure

The structure of the RNN we use is an all-inclusive Elman-Network [16], as delineated in figure 2. It

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

comprises of a forward feed part and a memory part; the last one stores the initiations of the feed-forward neurons from the past time step and fills in as extra contribution for the above part.
To play out the order, the time arrangement is nourished into the RNN and at the same time the enactment of the characterization neurons is watched.
This implies the quantity of arrangement neurons is equivalent to the quantity of classes N(c) and the characterization neuron with the most noteworthy yield esteem decides the order result. The arrangement of a period arrangement begins with the introduction of a warm-up grouping to the RNN so as to set the inward states effectively. From that point, the resulting time arrangement is sustained into the network and the yields y(c) of the order neurons are watched. In each time step, a champ takes these yields is performed, i. e., the order neuron with the most elevated initiation yields the characterization consequence of this single time step. Subsequent to displaying the entire time arrangement, the rates of wins hey of all the order neurons I ($1 \leq I \leq N(c)$) amid the whole time arrangement are assessed. The neuron with the most noteworthy rate determines the order result. The characterization mistake E (class) is given by the rate of misclassified input successions in an informational collection. The rates hey yield the arrangement result as well as give data about the unwavering quality r of the order result. In the event that the estimations of hello there are pretty much the equivalent for all characterization neurons, the order result is very flawed. Then again, on the off chance that one estimation of hello there overwhelms over all others, the RNN is very positive in its choice. 1 This thought can be evaluated by estimating the greatest estimation of hello . For standardization this esteem is mapped into the interim [0;1]. An increasingly complex meaning of r, which likewise considers the dissemination of the littler rates, can be given by methods for the entropy.
To yield an appropriate characterization result, the loads of the RNN must be enhanced regarding some execution measure. Managed learning (I. e., altering the association loads with the end goal that the correspondence between the network yield and wanted target information increments) gives a reasonable method for adjusting the loads and in this way the dynamic conduct of the RNN. Rather than enhancing the order mistake E (class) straightforwardly, the mean squared blunder is limited. To play out the arrangement effectively, reasonable elements speaking to the important qualities of the time arrangement must be developed in the RNN. As showed in area 1, we accept that the pertinent highlights are similar for related assignments. Also, such an extra undertaking presents an extra wellspring of data to develop a portrayal containing the principle highlights of the current issue. By adapting a well known performer Rprop [13] calculation, the Recurrent Neural Networks are prepared using iRprop +/- learning [9],[10]. The speculation execution of the prepared network is expanded by methods for cross-approval utilizing two arrangements of information amid preparing: the preparation informational collection is utilized to change the association loads. Those loads that produce the base E (class) on the approval informational index establish the last RNN arrangement. If there should be an occurrence of more than one network with a similar arrangement mistake, the network with the littlest mean squared blunder is picked. We use the test informational collection, a third informational collection that isn't utilized amid preparing, to decide the execution of the prepared RNN.
After training, the network is evaluated in order to find the accuracy, and thus the prediction of lifetime of the smartphone is done. In order to predict the lifetime, the inputs about the smartphone's features are given to the network, and the network produces an approximation of the lifetime of the product, which is compared to its actual lifetime, and thus accuracy is calculated. If this accuracy goes below a certain level, then the network is re-trained by changing the number of neurons, the number of layers, and accuracy is re-calculated. This process is repeated until we get the desired accuracy for the network. In the next section, we evaluate the results for the network, and compare it with existing algorithms like k-Nearest neighbor, Random forest classifier, MLP Classifier and Linear SVC classifier, the observations are also mentioned along with the comparisons.

IV. RESULT ANALYSIS AND CONCLUSION

We compared the accuracy of the developed RNN predictor with state of the art k-Nearest Neighbour, MLP, SVC and Random forest classifiers, and obtained the following results,

Table III. Comparison of Prediction Accuracy(%)

No. of Classfications	KNN	RF	MLP	SVC	RNN
5	60	60	60	56.25	80
10	70	70	70	65.63	80
20	75	70	72.5	67.97	85
25	76	72	74	69.38	84
30	73.33	73.33	73.33	68.75	83.33
45	68.89	77.78	73.33	68.75	82.22
50	70	80	75	70.31	85

From the above table, we can observe that the accuracy of the proposed RNN prediction system is better when compared to other techniques. We also observe that kNN is a good classifier for less number of samples, but as the number of samples increase, the accuracy of kNN reduces. The random forest classifier is a solid classifier, and maintains a good level of accuracy as the number of images increases, while the MLP & SVC classifiers gives moderate linearly increasing performance w.r.t. the number of images. These observations can be shown from the following figure,

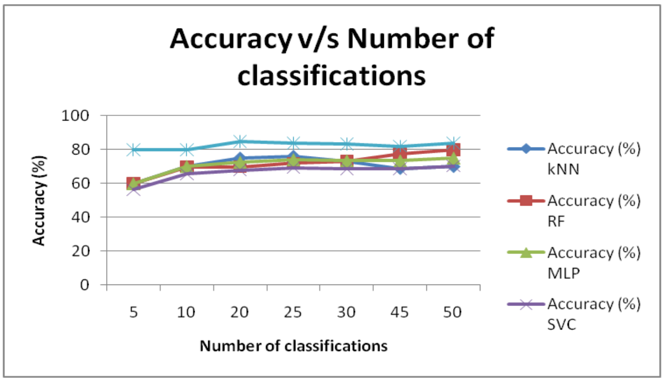

Figure 3. Accuracy Comparison

Thus, the RNN prediction system is a good model for prediction of lifetime of the products, and can be used for real time applications like smart phone obsolescence detection.

V. FUTURE WORK

The existing work gives a good enough approximation about the accuracy of lifetime prediction for the smartphone domain. Researchers can further extend this work, and add techniques like machine learning and artificial intelligence which can adapt to the learning rate based on the product's behaviour and real time market analysis, which will further tune the accuracy of prediction of lifetime for the product.

REFERENCES

1. S. Lawrence, C. L. Giles, and S. Fong. Natural language grammatical inference with recurrent neural networks. IEEE Transactions on Knowledge and Data Engineering, 12(1):126–140, 2000.
2. M. Riedmiller. Advanced supervised learning in multi-layer perceptrons – from backpropagation to adaptive learning algorithms. International Journal of Computer Standards and Interfaces, 16(5):265–278, k4.
3. G.-Z. Sun, H.-H. Chen, and Y.-C. Lee. Green's function methods for fast on-line learning algorithm of recurrent neural networks. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, Advances in Neural Information Processing 4, pages 333–340, San Mateo, CA, k2. Morgan Kaufmann Publishers.
4. R. I. Watrous and G. M. Kuhn. Induction of finite-state automata using second-order recurrent neural networks. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, Advances in Neural Information Processing Systems 4, pages 317–324, San Mateo, CA, k2. Morgan Kaufmann Publishers.
5. P. Stagge and B. Sendhoff. An extended Elman net for modeling time series. In W. Gerstner, A. Germond, M. Hasler, and J. Nicoud, editors, Artificial Neural Networks (ICANN 97), volume 1327 of Lecture Notes in Computer Science, pages 427–432, Berlin, k7. Springer Verlag.
6. J. T. Connor, D. Martin, and L. E. Atlas. Recurrent neural networks and robust time series prediction. IEEE Transactions on Neural Networks, 5(2):240–254, 2004.
7. C. Igel and M. Husk ¨ en. Improving the Rprop learning algorithm. In H.-H. Bothe and R. Rojas, editors, Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), pages 115–121, Canada / Switzerland, 2000. ICSC Academic Press.
8. E. W. Saad, D. V. Prokhorov, and D. C. Wunsch. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE Transactions on Neural Networks, 9(6):1456–1470, k8.
9. R. Caruana. Multitask learning. Machine Learning, 28:41–75, 2017.
10. P. Stagge and B. Sendhoff. Organisation of past states in recurrent neural networks: Implicit embedding. In M. Mohammadian, editor, Computational Intelligence for Modelling, Control & Automation, pages 21–27, Amsterdam, k9. IOS Press.
11. K. Doya. Recurrent networks: Supervised learning. In M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks, pages 796–800. The MIT Press, Cambridge, MA, 2008.
12. C. Igel and M. Husk ¨ en. Empirical evaluation of the improved Rprop learning algorithm. Neurocomputing, 50(C):105–123, 2003.
13. C. L. Giles, C. B. Miller, D. Chen, G. Z. Sun, H. H. Chen, and Y. C. Lee. Extracting and learning an unknown grammar with recurrent neural networks. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, Advances in Neural Information Processing Systems 4, pages 317–324, San Mateo, CA, k2. Morgan Kaufmann Publishers.
14. C. W. Omlin and C. L. Giles. Extraction of rules from discrete-time recurrent neural networks. Neural Networks, 9(1):41–52, k6.
15. A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland. Finite state automata and simple recurrent networks. Neural Computation, 1:372–381, 2009.
16. M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the IEEE International Conference on Neural Networks, pages 586–591, New York, k3. IEEE Press.

AUTHORS PROFILE

Manasvi Gurnaney completed her Bachelor of Engineering in Computer Science from Shri Ramdeobaba College of Engineering and Management, Nagpur, India in 2015 and currently pursuing P.G in Computer Science and Engineering from Shri Ramdeobaba College of Engineering and Management, Nagpur, India. She has 2 years of Industry Experience. The Research Paper work focuses on Data Mining and Machine Leaning.

Dr. Shubhangi Neware completed her bachelors degree (B.E.) in Computer Technology from Nagpur University in year 1999 and Masters Degree (M.Tech) in Computer Science and Engineering from Chhattisgarh Swami Vivekanand Technical University in year 2009. She has completed her PhD on topic "Finger Knuckle Surface based Biometrics for Personal Identification" in year 2017 from CVRU, Bilaspur. She has 16 years of teaching experience and 2 years of Industry Experience.She is currently working as Assistant Professor in the Department of Computer Science & Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur. She has published more than 20 research papers in reputed international journals and conferences. Her area of interest includes Image Processing , Biometrics, Pattern Recognition and Data mining.