# Production enhancement by prediction of liquid steel breakout in continuous casting process

Md Obaidullah Ansari[1]*, Somnath Chattopadhyaya[1], Joyjeet Ghosh[1], and Priyansh Sharma[2]

[1] Indian Institute of Technology, Dhanbad, India
{mdobaidullahdps,somuismu,joyjeetghose}@gmail.com
[2] Birla Institute of Technology Mesra, Ranchi, India

**Abstract.** Breakout is one of the major accidents in continuous casting shops that affect the temporary shutdown of slab caster, damage of machinery due to splash of molten steel, capital loss, affecting productivity and safety hazards. At Bokaro Steel Plant, the existing breakout prediction system is based on a logical judgement algorithm controlled by PLC to predict breakout and generate alarm, highly dependent on the operator experience. This paper presents an Artificial Neural Network with Backpropagation Algorithm which has been developed using Python language to write the code and used Keras framework and Tensor-Flow as a backend and it uses Adam optimizer and Binary cross-entropy loss function to predict the liquid breakout in caster and also avoid the operator intervention. The experimental results show that the model has 100% accuracy to generate an alarm for true breakout and reduce the false alarm upto 0%

**Keywords:** Continuous Casting, Mould Breakout, Artificial Neural Network, .

## 1 Introduction

Continuous casting of steel is a process in which liquid steel is continuously solidified into a strand of metal [1]. Depending on the dimensions of the strand, these semi-finished products are called slabs, blooms or billets. Presently, most of the steel manufacturing industries worldwide are using a continuous casting process and more than 90% of the steel is produced by continuous casting process [2-3]. Mould is the heart of the caster where the solidification process starts [4-5]. A little difference may affect the productivity or quality of cast slabs. At Bokaro Steel Plant, straight mould is used and is made of copper, with 900 mm length, thickness 200–225 mm & 2000 mm width as shown in figure 1.

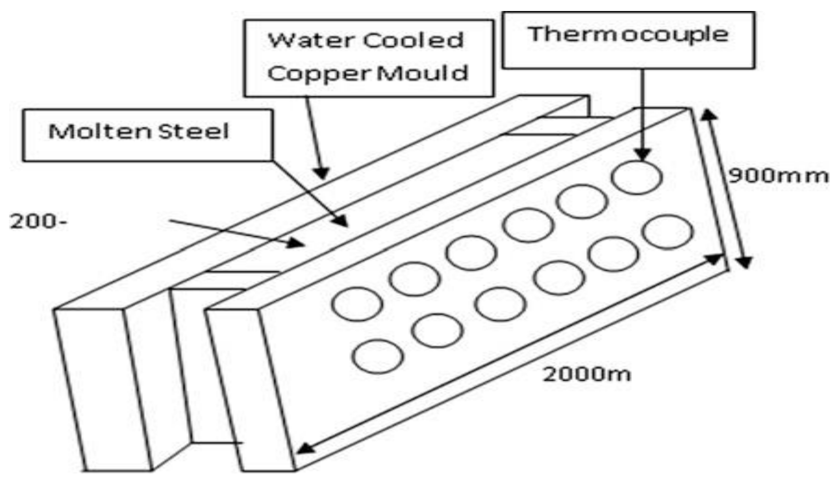


Figure 1: Copper mould

There are lots of problems in continuous casting shops but a major problem is mould breakout of liquid steel. The molten steel in the mould is gradually cooled and solidified to be drawn out from a lower portion of the mould as a strand. When the molten steel is cooled within the mould, a solidified portion is called a shell which is formed on the surface of the molten steel. The shell sometimes sticks to the mould plate and often cracks due to various factors. When a crack portion of the shell reaches the bottom of the mould, the molten steel in the shell leaks out from the mould. Such an occurrence is called breakout as shown in figure 2.

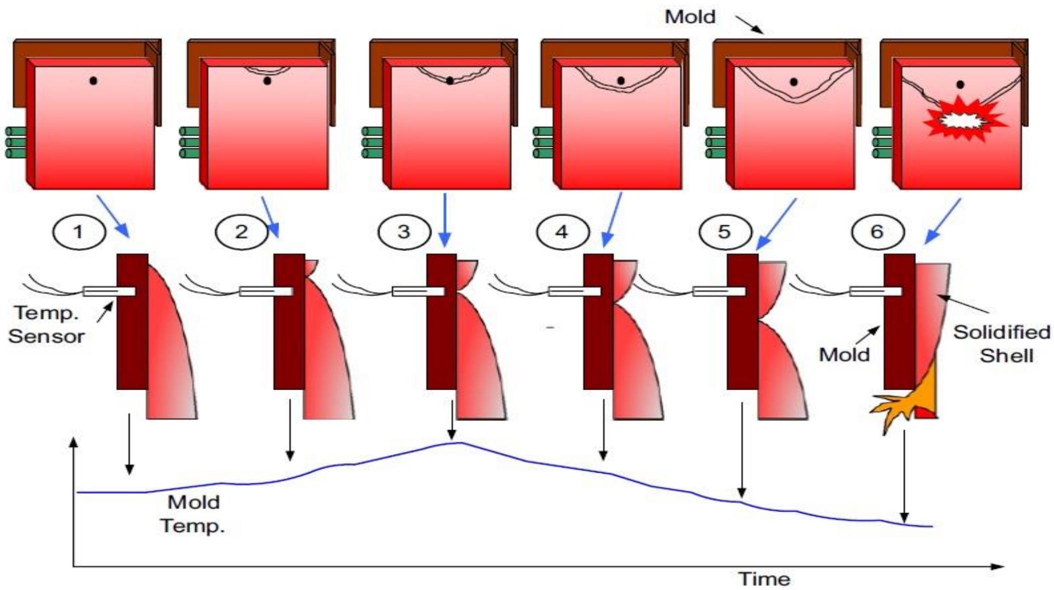


Figure 2: A schematic diagram of breakout.

Breakout leads to temporary shutdown of slab caster, damage of machinery due to splash of molten steel, capital loss, affecting productivity and safety hazards [6-7].

## 2 LOSSES DUE TO BREAKOUT:

After breakout an average 3 to 4 hours required to restart the Caster. Average delay due to breakout = (6×4×12) = 288hours/years. The total capital loss caused by the loss of liquid steel per year is 86 crores as shown in Table 1.

Table 1: Loss Due Breakout

| Particular | Unit | Value |
|---|---|---|
| Number of Breakout occur due sticker | Number per month | 06 |
| After breakout an average 3 to 4 hours required to restart the Caster. Average delay due to breakout = (6×4×12) = **288hours/years** | | |
| Average weight of one heat | Ton | 280 |
| Cost of one ton of steel | INR | 42000 |
| Loss of liquid steel per year | INR | **86 crore/year** |

Breakout can be occurred due to many reasons, but after collected last three years data from operational log books of Bokaro steel plant, it is clear that the 90% of the breakout is due to first four problems i.e., sticker, taper/mould, casting speed and mould level [8-10]. More than 70-80 % breakout occurred due to sticker breakout [11-12]. Mainly there are two types of breakout prediction system based on thermocouple temperature: logical judgement method [13-17] and artificial intelligent method [18-21]. In Bokaro Steel Plant, a logical based breakout prediction system is used to predict the breakout by using variation in multiple thermocouple temperatures. Logical judgement-based systems are totally depending on thermocouple temperature, caster equipment, casting speed, mould friction etc. [22-23] and generate false alarms or even fail to generate alarms before breakout.

Literature survey shows that different types of artificial neural network [2,6,24-26], Support Vector Regression [27] and K-means clustering [22,28] has been used to predict the breakout by using only thermocouple temperature or mould friction as an input but breakout also depends on casting speed and mould level.

**Breakout prediction model**

In this paper, an Artificial intelligent based breakout prediction model was developed, the flowchart is shown in figure 3. First, the data is collected from a steel melting shop (SMS-II) of Bokaro steel Plant. EDA (exploratory data analysis) technique is used for better understanding of data, their relationships and patterns by correlation matrix, pairplot and histogram. After this the data is standardised and then it is trained through the Neural Network which is best suited for it. The neural network selected is feed-forward neural network (also known as artificial neural network), along with backpropagation which minimizes the loss by backtracking algorithm. After getting the desired loss, the model is selected and then tested on the test data and the accuracy is measured.

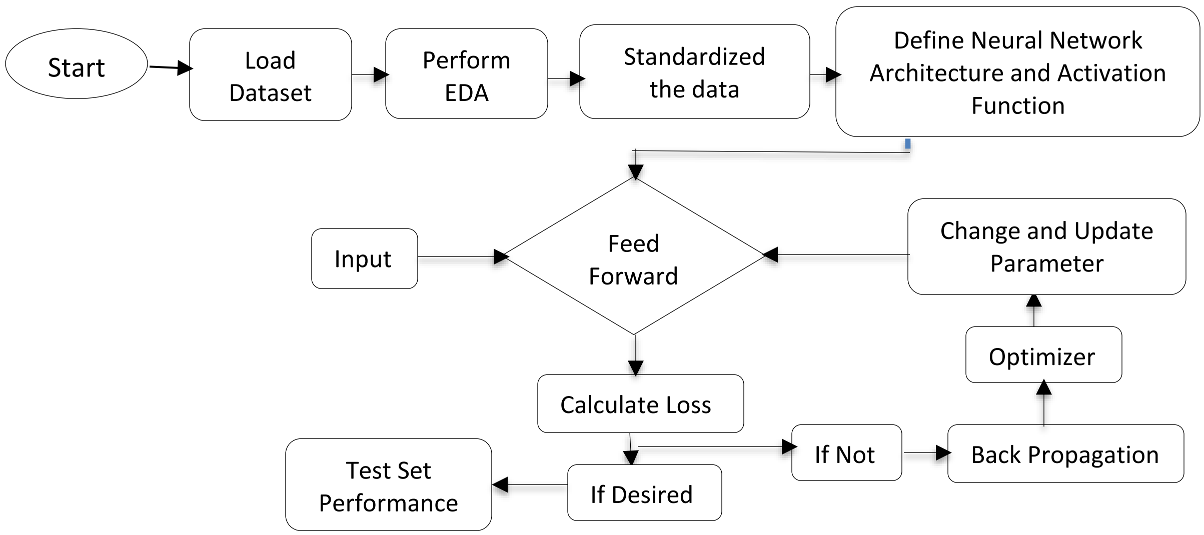


Figure 3: Flowchart

In this work, an Artificial Neural Network with Backpropagation Algorithm has been developed by using Python language to write the code and using Keras framework and Tensor-Flow as a backend. The optimizer used is Adam optimizer and the loss function is Binary cross-entropy loss function. 180 different observations were collected from the operational log book of Bokaro Steel Plant's continuous casting shop at the time of breakout and no breakout to form the dataset, which contain 10 different features out of which 8 were the thermocouple temperatures (T1, T2 ….T8) and other two were casting speed (CS) and the mould level (ML). The outputs are either 0 (for no breakout) or 1 (for breakout). When there was a breakout the alarm was triggered.

## 3 Exploratory Data Analysis (EDA)

EDA is a method of describing the data by means of statistical and visualization techniques so as to better understand the data, their relationships and patterns. Below are some of the visualizations of the data.

### 3.1 Correlation Matrix

The correlation between all the variables of the dataset is found using a correlation matrix as shown in the figure 4. Each row and column represent a variable, and each value in this matrix is the correlation coefficient between the variables represented by the corresponding row and column. Values nearing +1 indicate the presence of a strong positive relationship between the two variables, whereas those nearing -1 indicate a strong negative relation between those two variables. Values near to zero mean there is no relationship between the two variables.

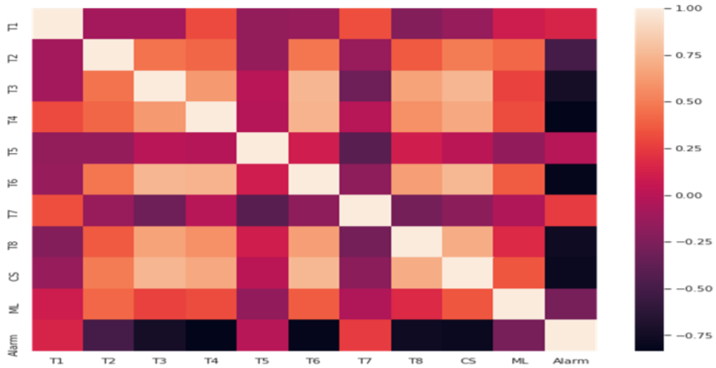

Fig.4: Correlation Matrix

From the above figure 3, it is deduced that the relationship of alarm with T7 and T1 is positive and with CS, T8, T6, T4, T3, T2 and ML is negative and with T5 is almost equal to zero.

### 3.2 Pairplot

A pairplot plots a pairwise relationship in a dataset. The pairplot function creates a grid of axes such that each variable in data is shared in the y-axis across a single row and in the x-axis across a single column.

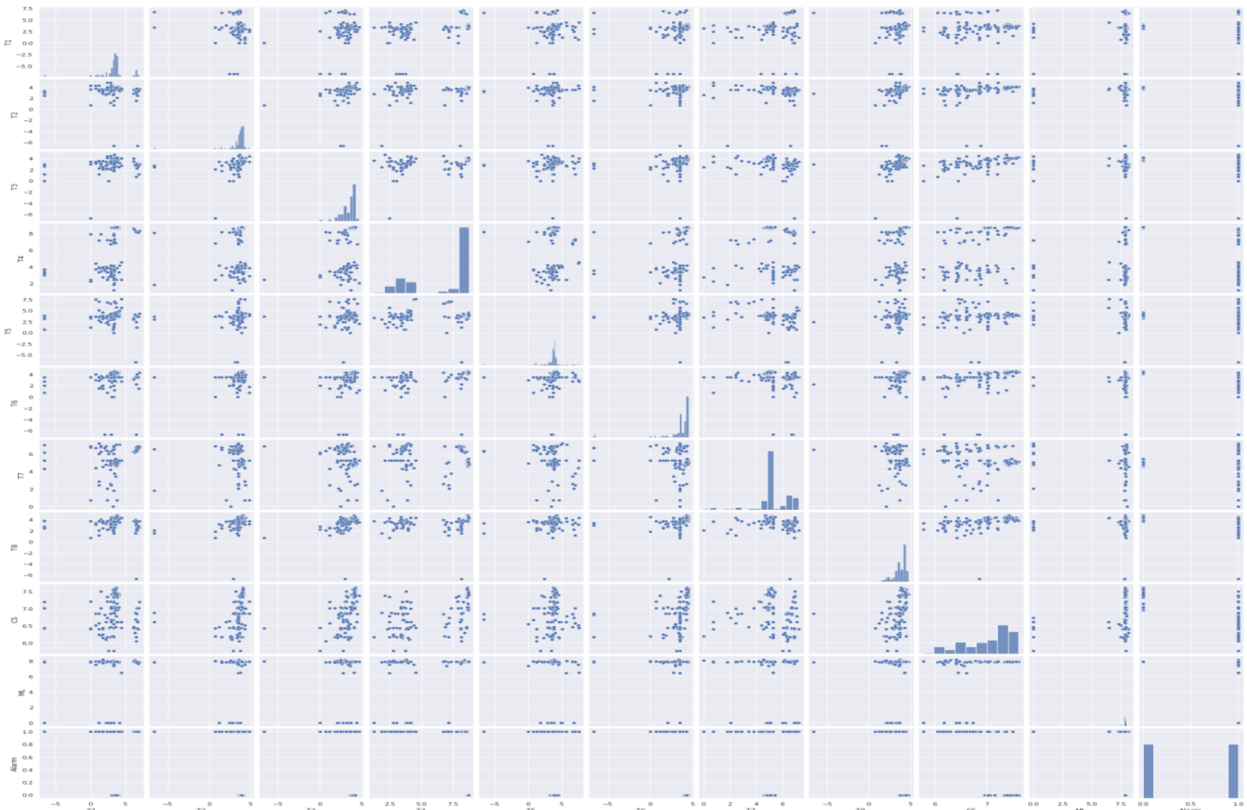

Figure 5: Pairplot

### 3.3 Histogram

It represents the frequency distribution of all the features.

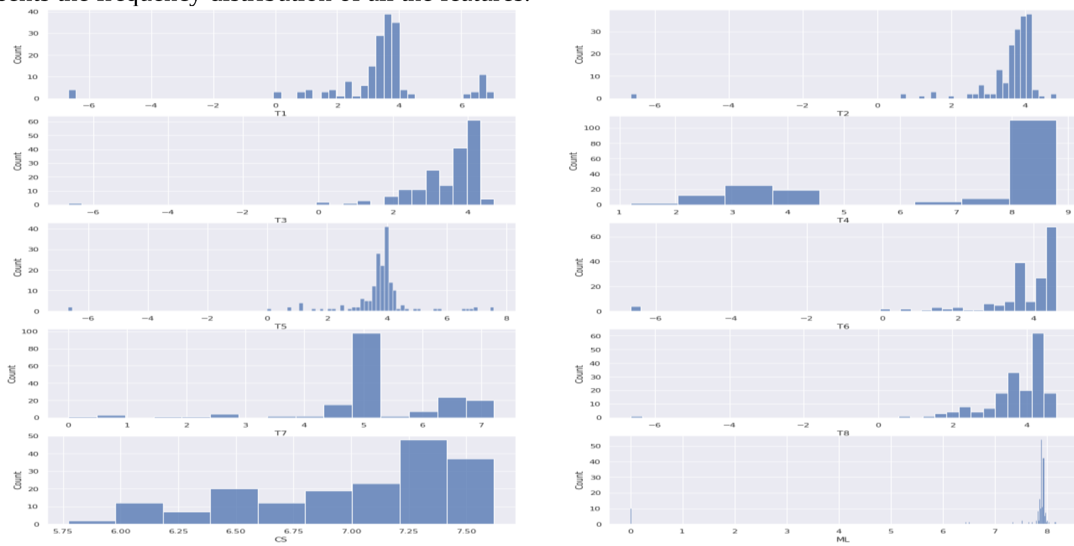


Figure 6: Histogram

From the pairplot (figure 5) and histogram plots (figure 6) it can inferred that the distribution does not follow normal distribution and the data is slightly skewed but as the dataset is quite smaller, a small amount also cannot be afforded to lose, while removing the skewness through any transformation method (like the box-cox transformation). Therefore, the data will only be standardized.

## 4 Data Preprocessing

It is the process of making the raw data suitable for the training purpose.

### 4.1 Splitting the dataset into train and test data

Before standardizing the data, the dataset is splitted into train and test sets respectively using the train_test_split function of the sklearn library, so that the train and test sets are standardized separately so as to obtain better accuracy.

### 4.2 Standardizing the data

Standardizing the data means transforming the data such that its distribution has a mean value is 0 and a standard deviation is 1. To achieve this, the mean value is subtracted from each value of the dataset and then divide it by the standard deviation of the whole dataset. Since our dataset has multivariate data, this is done feature-wise i.e., for each column independently and use the standard scaler from the sklearn. The formula is given below.

$$Standardization\ z = \frac{x - \mu}{\sigma}$$

$$with\ mean: \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$$

$$and\ standard\ deviation: \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

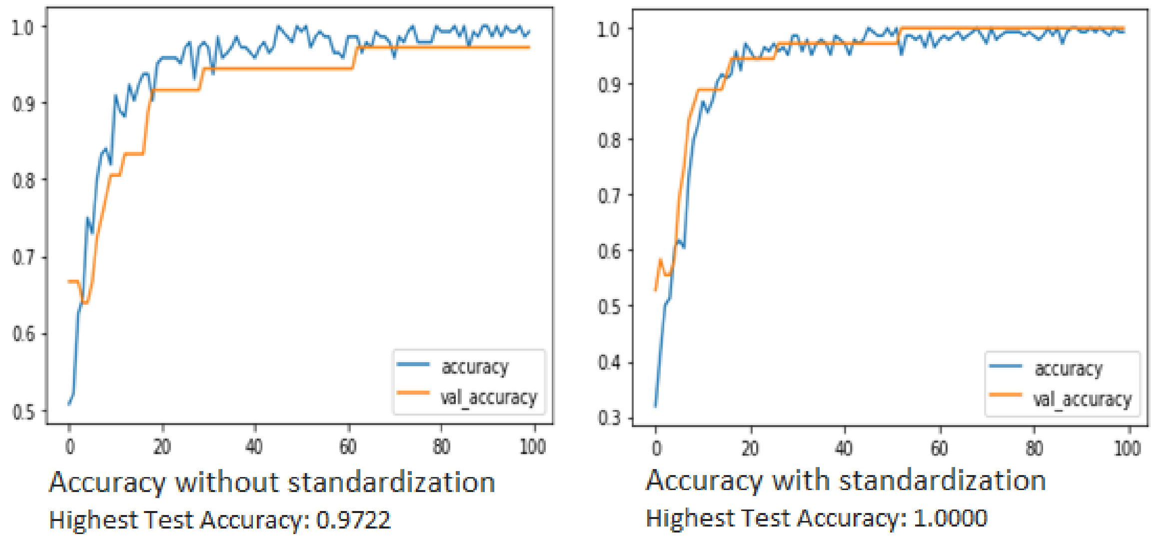


Figure 7: Accuracy cure with and without standardization

The comparison between the accuracy curves of the model without and with standardization is given below. Accuracy curve shows that model with standardized data test accuracy 1.0000 as compared to without i.e., 0.9722 as shown in above figure 7. After standardization data is ready to get trained.

## 5 Neural Network Architecture

### 5.1 Hidden and output layers and Activation function

In Artificial neural networks, the first layer is called the input layer, it consists of the 10 inputs which are the different features of every observation of the dataset. The last layer is called the output layer, it contains 1 neuron as it is a binary classification task, and the layers present in between the input and the output layer are called the hidden layers. In this Neural Network, there are two hidden layers, the first hidden layer having 16 neurons and the second hidden layer having 12 neurons. For the first and second hidden layer, Rectified linear unit (ReLu) activation function and for the Output layer, sigmoid activation function has been used.

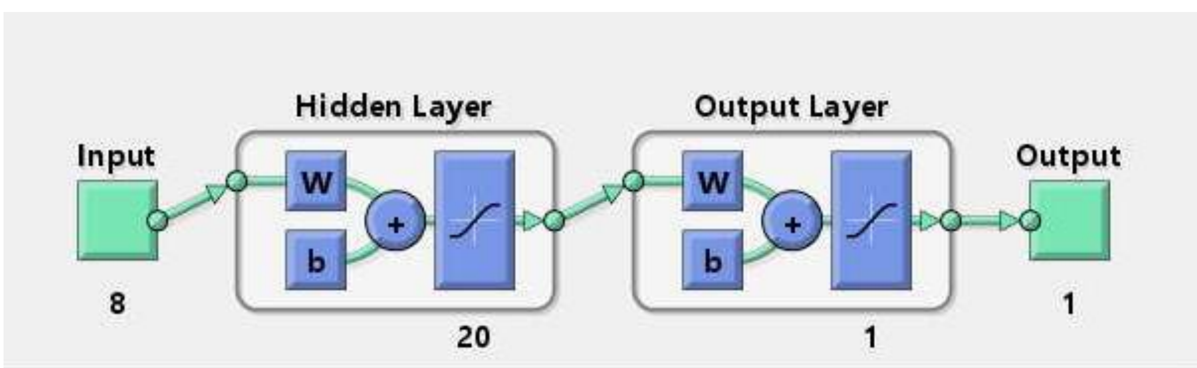


Figure 8: Architecture of Neural Network

The sigmoid activation function is used majorly in the output layers and is used for predicting probability-based outputs and has been successfully implemented in binary classification problems while the ReLu function is the most widely used activation function. It usually achieves better performance and generalization in deep learning compared to the sigmoid activation function. Therefore, the sigmoid function is especially used for the output layer and ReLu function is used for other 2 hidden layers for better performance.

As sigmoid function is used for the output layer the predicted probability of mould breakout which will be between 0 and 1 is calculated by the ANN. For the final values these probabilities have to be converted to either 0 or 1 depending on the threshold value which is 0.5 by default.

The graph of Sigmoid and ReLu functions are shown in figure 9

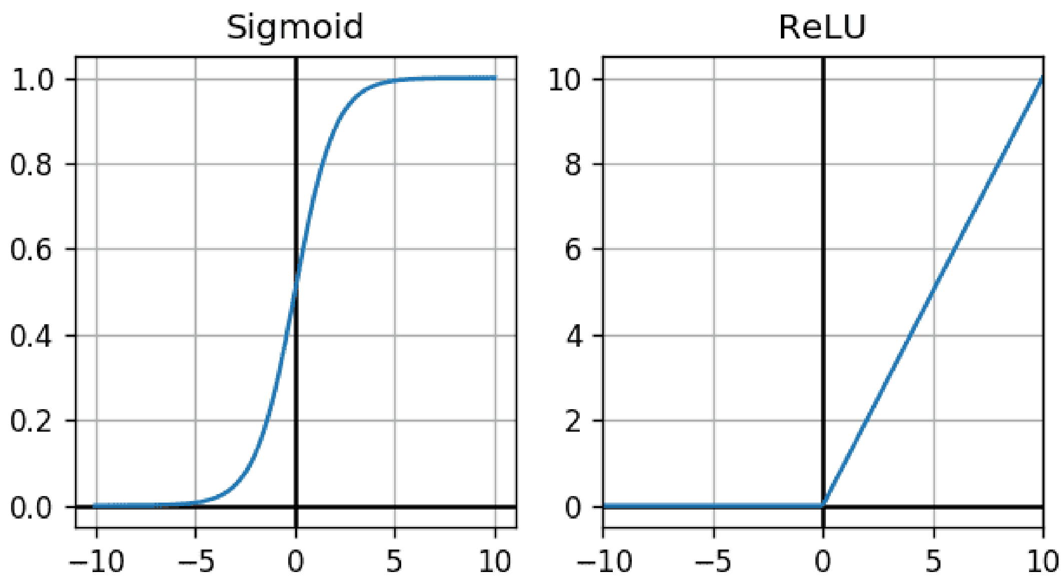


Figure 9: Graph of sigmoid and ReLu function

**Sigmoid Function**

$\Phi(z) = 1/(1+\exp(-z))$

**ReLu Function**

$f(x) = \max(0,x)$

Which means $f(x) = 0$ if $x<=0$ or else $f(x) = x$ if $x>0$

**5.2 Dropout and Batch-Normalization**

For improving the model performance, Dropout and Batch-Normalization technique has been used after each of the two hidden layers. Dropout technique is used to prevent neural network models from overfitting. The dropout parameter is set 0.3 for this model [29]. The working of dropout in general is shown in figure 10.

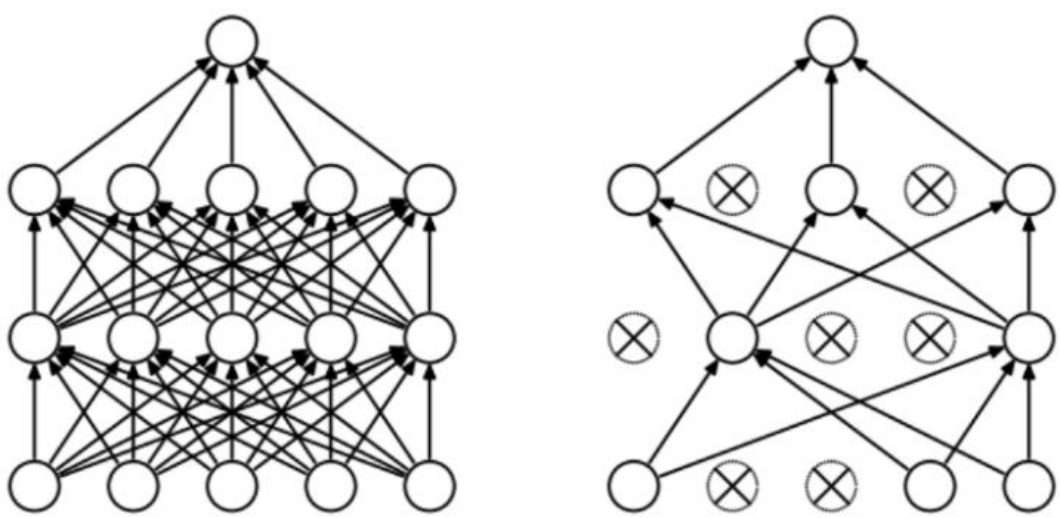

Figure 10: Standard Neural Network and after applying dropout.

Batch-Normalization is a technique for training neural networks that standardizes the inputs to a layer for each mini-batch [30]. This has the effect of stabilizing the learning process and reducing the number of training epochs required to train deep networks. The formula used for batch normalization is shown below.

The forward propagation of Batch-Norm is shown below:

$\text{Input: Value of x over a mini} - \text{batch: } \beta = \{x_1 \ldots m\};$
$\text{Parameters to be learned: } \gamma, \beta$
$\text{Output: } \{BN_{\gamma,\beta}(x_i)\}$

$$\mu_\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad //\text{mini} - \text{batch mean}$$

$$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\beta)^2 \qquad //\text{mini} - \text{batch veriance}$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \in}} \qquad //\text{normalize}$$

$$y_i \leftarrow \gamma\widehat{x_i} + \beta \equiv NB_{\gamma,\beta}(x_i) \qquad //\text{scale and shift}$$

The comparison between the accuracy curves of the model without and with dropout and batch normalization is given below in figure 11.

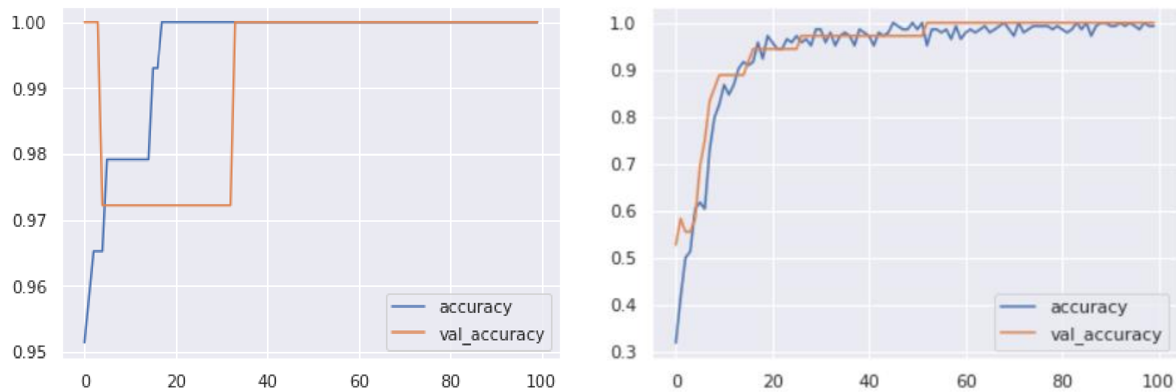


Figure 11: Accuracy curve of the model without and with dropout and batch normalization

From the accuracy curve of the model without dropout and batch normalization it can be deduced that the model is overfitted if dropout and batch normalization is not used.

**5.3 Selection of Optimizer and Loss Function**

Optimizers are the algorithms which are used to adjust the attributes of neural networks such as weights and learning rates in order to reduce the losses during the training process.

Adam optimizer is used as it combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems [31]. Adam is relatively easy to configure where the default configuration parameters do well on most problems. It is therefore efficient and comparatively faster. Since our problem is a classification problem therefore, binary cross-entropy loss function is used. In earlier research papers MSE (Mean squared error loss) was used but it is used for regression tasks not for classification tasks, because the decision boundary in a classification task is large (in comparison with regression). Setting the epochs to 100 and the batch size as 16 after implementing the above model we are getting a maximum of 100% accuracy in our both training and testing sets. The binary cross-entropy loss function calculates the loss of an example by computing the following average:

$$Loss = -\frac{1}{output\ size}\sum_{i=1}^{output\ size} y_i.\log y_i + (1 - y_i).\log(1 - \hat{y}_i)$$

where $\hat{y}_i$ is the ith scalar value in the model output, $y_i$ is the crossesponding target value and outpit size is the number of scalar value in the model output

**5.4 Accuracy Curve**

It can be seen that the accuracy is increasing as we are increasing the number of epochs. Both the train and test accuracy are becoming 100% as shown in figure 12.

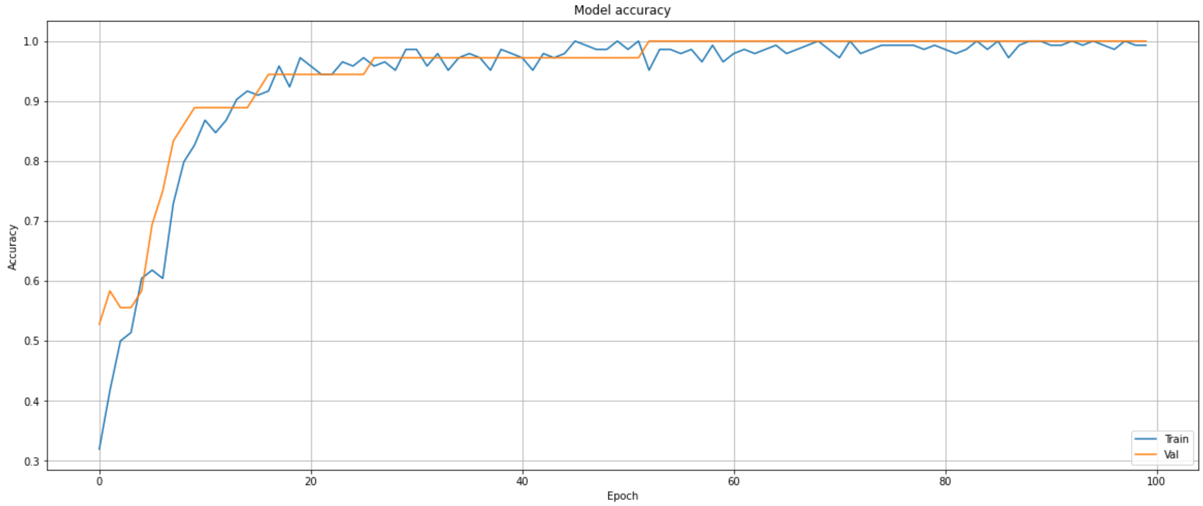


Figure 12: Accuracy curve of the model

**5.5 Loss curve**

The loss is decreasing with the number of epochs as shown in figure 13.

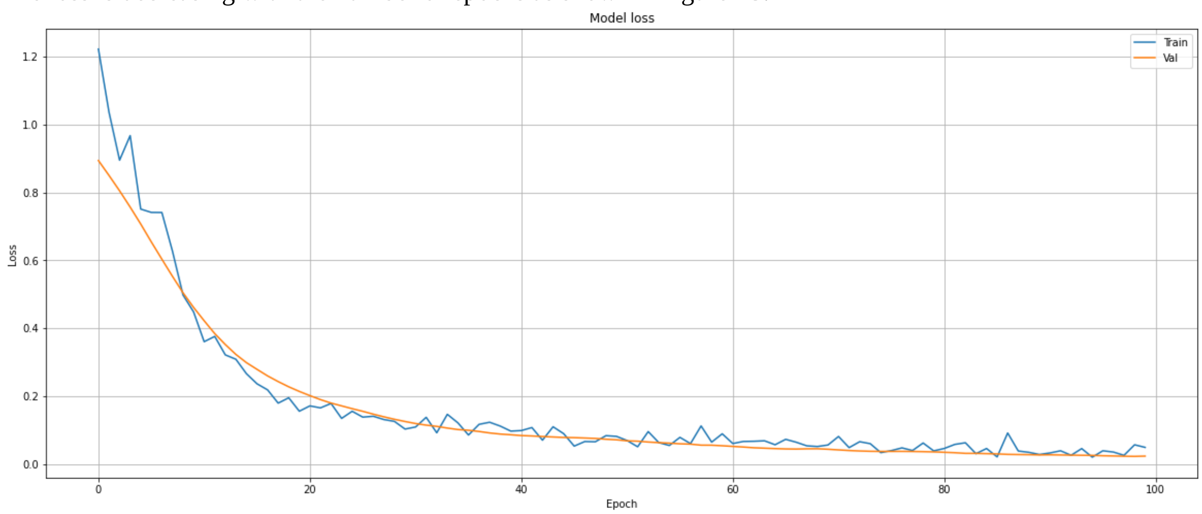


Figure 13: Loss Curve of the Model

| | |
|---|---|
| Minimum train loss: 0.0204 | Minimum test Loss: 0.0227 |
| Highest Train Accuracy: 1.0000 | Highest test Accuracy: 1.0000 |

From the loss curve, it is clear that loss is decreases with number of epochs and model having minimum train loss (0.0204) and minimum test loss (0.0227). Accuracy curve indicate that both train and test accuracy are become 100% with highest train accuracy (1.0000) and highest test accuracy (1.0000). Testing result shows that model is accurately predict that breakout.

## 6 Summary Table of the Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_9 (Dense)              (None, 16)                176
_________________________________________________________________
batch_normalization_6 (Batch (None, 16)                64
_________________________________________________________________
dropout_6 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_10 (Dense)             (None, 12)                204
_________________________________________________________________
batch_normalization_7 (Batch (None, 12)                48
_________________________________________________________________
dropout_7 (Dropout)          (None, 12)                0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 13
=================================================================
Total params: 505
```

Now developed a system that will automatically control the casting speed, whenever BOPS alarms or teeming interrupt commands are generated.

## 7 DEVELOPMENT OF FRAMEWORK FOR AUTOMATIC REDUCTION OF CASTING SPEED

Now proposed a model that will automatically control the casting speed whenever there is a breakout alarm or teeming interrupt (when liquid steel level is increased within the mould then the tundish slide gate is closed automatically and vice-versa. Normal opening of the tundish slide gate is maximum 78% and minimum is 48%. Whenever this range is crossed by a tundish slide gate, an interrupt command is generated by automatic mould level control (AMLC)) is generated. This model also controls the casting according to the grade of steel. When teeming interrupt is generated then casting speed is reduced to creeps speed i.e. 0.1m/min. In case of breakout alarm casting speed will reduce to 0.8m/min as shown figure 13.

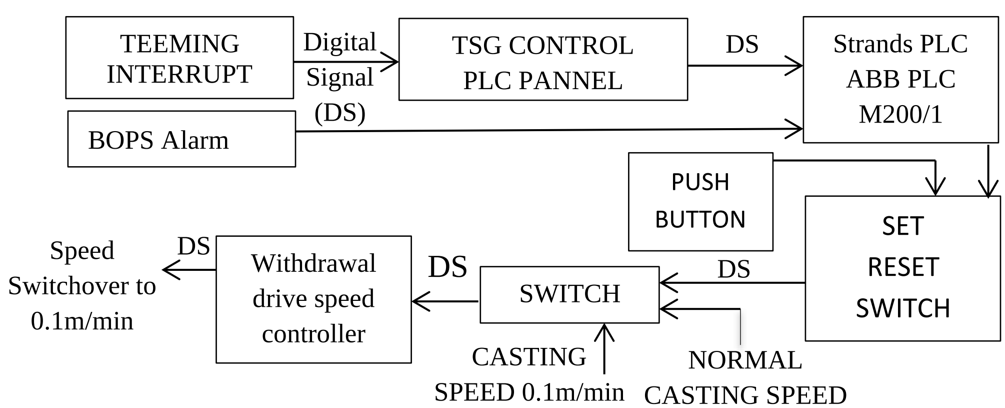


Figure 13: Proposed modem to reduce casting speed automatically

In this proposed model both breakout alarm and teeming interrupt are considered to control the casting speed. Whenever teeming interrupt is generated at the same time TSG (tundish slide gate) control sends a command to the strands PLC (programmable logic controller). This strands PLC control the casting with the help of a withdrawal drive speed controller. In case of teeming interrupt casting speed, we'll reduce to 0.1m/min. At the time of breakout alarm, strands PLC well generates a signal to control the casting speed with the help of withdrawal drive. In this case casting speed well reduces to 0.8m/min.

## Conclusion:

In this work, an artificial neural network with backpropagation mold breakout prediction has been developed by using Keras framework and Tensor-Flow as a backend. In previous work, breakout prediction system has developed by using thermocouple temperature or mould frection. In this system, thermocouple temperature, mould level and casting speed are used to predict the breakout. Accuracy curve and testing result show that this system successfully predicts all types of breakout and even reduce to generate false alarm during casting as compare to the existing BOPS system in Bokaro steel plant. At the time of breakout alarm, casting speed will reduce to 0.8m/min and in case teeming interrupt casting speed is reduce to creeps speed i.e. 0.1m/min

## References:


1. Pan, Ershun, Liang Ye, Jianjun Shi, and Tzyy-Shuh Chang. "On-line bleeds detection in continuous casting processes using engineering-driven rule-based algorithm." *Journal of manufacturing science and engineering* 131, no. 6 (2009).
2. Hore, Sheuli, Suchandan K. Das, Manoj M. Humane, and Anil Kumar Peethala. "Neural Network Modelling to Characterize Steel Continuous Casting Process Parameters and Prediction of Casting Defects." *Transactions of the Indian Institute of Metals* 72, no. 12 (2019): 3015-3025.
3. Louhenkilpi S. Continuous casting of steel. In Treatise on process metallurgy 2014 Jan 1 (pp. 373-434). Elsevier.
4. Qian, Haitao, Fei He, Xi’an Xie, Zhenghai Zhu, Lingying Zhang, Linlin Zhang, and Jiashun Shi. "Recovery mechanism of the sticking-type breakout during continuous casting of steel." *Ironmaking & Steelmaking* 46, no. 3 (2019): 259-268.
5. Wang, Wanlin, Chenyang Zhu, and Lejun Zhou. "Initial Solidification and Its Related Heat Transfer Phenomena in the Continuous Casting Mold." *steel research international* 88.10 (2017): 1600488.
6. He, Fei, and Lingying Zhang. "Mold breakout prediction in slab continuous casting based on combined method of GA-BP neural network and logic rules." *The International Journal of Advanced Manufacturing Technology* 95, no. 9 (2018): 4081-4089.
7. Duan, Haiyang, Xudong Wang, Yu Bai, Man Yao, Yu Liu, and Qingtao Guo. "Modeling of breakout prediction approach integrating feature dimension reduction with K-means clustering for slab continuous casting." *The International Journal of Advanced Manufacturing Technology* 109, no. 9 (2020): 2707-2718.
8. He, Fei, and Li Zhou. "Formation Mechanism and Influence Factors of the Sticker between Solidified Shell and Mold in Continuous Casting of Steel." *High Temperature Materials and Processes* 38, no. 2019 (2019): 192-198.
9. Salah, Bouhouche, Mentouri Zoheir, Ziani Slimane, and Bast Jurgen. "Inferential sensor-based adaptive principal components analysis of mould bath level for breakout defect detection and evaluation in continuous casting." *Applied Soft Computing* 34 (2015): 120-128.
10. Li, Wentao, Yang Li, and Yuan Zhang. "Study of mould breakout prediction technique in continuous casting production." In *2010 3rd International Conference on Biomedical Engineering and Informatics*, vol. 7, pp. 2966-2970. IEEE, 2010.
11. Luk’yanov, S. I., E. S. Suspitsyn, S. S. Krasilnikov, and D. V. Shvidchenko. "Intelligent system for prediction of liquid metal breakouts under a mold of slab continuous casting machines." *The International Journal of Advanced Manufacturing Technology* 79, no. 9 (2015): 1861-1868.
12. He, Fei, Li Zhou, and Zhi-hao Deng. "Novel mold breakout prediction and control technology in slab continuous casting." *Journal of Process Control* 29 (2015): 1-10.
13. He, Fei, D-F. He, Z-H. Deng, A-J. Xu, and N-Y. Tian. "Development and application of mould breakout prediction system with online thermal map for steel continuous casting." *Ironmaking & Steelmaking* 42, no. 3 (2015): 194-208.
14. Watzinger, J., A. Pesek, N. Huebner, M. Pillwax, and O. Lang. "MoldExpert–operational experience and future development." *Ironmaking & steelmaking* 32, no. 3 (2005): 208-212.
15. Qin, Xu, Chao-Fu Zhu, Yan-Rong Yin, and Xi-Rong Dong. "Forecasting of molten steel breakouts for the slab continuous casters with hydraulic servo oscillation systems." *Iron and Steel/ Gangtie* 45, no. 11 (2010): 97-100.
16. Qin, Xu, Chaofu Zhu, and Liwen Zheng. "Study of the forecasting of molten steel breakouts based on the frictional force between mould and slab shell." In *2010 International Conference on Mechanic Automation and Control Engineering*, pp. 2593-2596. IEEE, 2010.

17. Bhattacharya, Arya K., P. S. Srinivas, K. Chithra, S. V. Jatla, and Jadav Das. "Recognition of fault signature patterns using fuzzy logic for prevention of breakdowns in steel continuous casting process." In *International Conference on Pattern Recognition and Machine Intelligence*, pp. 318-324. Springer, Berlin, Heidelberg, 2005.
18. Liu, Yu, Xudong Wang, Fengming Du, Man Yao, Yali Gao, Fuwang Wang, and Junyao Wang. "Computer vision detection of mold breakout in slab continuous casting using an optimized neural network." *The International Journal of Advanced Manufacturing Technology* 88, no. 1-4 (2017): 557-564.
19. Cheng, Ji, Cai Zhao-Zhen, Tao Nai-Biao, Yang Ji-Lin, and Zhu Miao-Yong. "Molten steel breakout prediction based on genetic algorithm and BP neural network in continuous casting process." In *Proceedings of the 31st Chinese Control Conference*, pp. 3402-3406. IEEE, 2012.
20. Zhang, Ben-guo, Qiang Li, Ge Wang, and Ying Gao. "Breakout prediction based on BP neural network of LM algorithm in continuous casting process." In *2010 International Conference on Measuring Technology and Mechatronics Automation*, vol. 1, pp. 765-768. IEEE, 2010.
21. Xu, Zening, Lianjun Ma, Yongguo Zhang, Xu Qin, and Yuanzhe Cui. "Study on neural network breakout prediction system based on temperature unit input." In *2010 International Conference on Measuring Technology and Mechatronics Automation*, vol. 3, pp. 612-615. IEEE, 2010.
22. Duan, Haiyang, Xudong Wang, Yu Bai, Man Yao, and Qingtao Guo. "Integrated approach to density-based spatial clustering of applications with noise and dynamic time warping for breakout prediction in slab continuous casting." *Metallurgical and Materials Transactions B* 50, no. 5 (2019): 2343-2353.
23. Xu, Qin, Chao-fu Zhu, Li-wen Zheng, and Yan-rong Yin. "Molten steel breakout prediction based on thermal friction measurement." *Journal of Iron and Steel Research, International* 18, no. 4 (2011): 24-35.
24. Ren, Tingzhi, Xueliang Shi, Dongmei Li, Xin Jin, Yujuan Wu, and Weiqiang Sun. "Research on breakout prediction system based on multilevel neural network." In *2010 International Conference on Electrical and Control Engineering*, pp. 1652-1655. IEEE, 2010.
25. Liu, Kai, Li Peng, and Qin Yang. "The algorithm and application of quantum wavelet neural networks." In *2010 Chinese Control and Decision Conference*, pp. 2941-2945. IEEE, 2010.
26. Faizullin, Artur, Mikhail Zymbler, Dirk Lieftucht, and Felix Fanghänel. "Use of deep learning for sticker detection during continuous casting." In *2018 Global Smart Industry Conference (GloSIC)*, pp. 1-6. IEEE, 2018.
27. Qing, Tian, Wang Jia-Wei, and Xue Ji-Shuang. "Research for breakout prediction system based on support vector regression." In *2012 IEEE Symposium on Robotics and Applications (ISRA)*, pp. 188-190. IEEE, 2012.
28. Duan, Haiyang, Xudong Wang, Yu Bai, Man Yao, and Qingtao Guo. "Application of k-means clustering for temperature timing characteristics in breakout prediction during continuous casting." *The International Journal of Advanced Manufacturing Technology* 106, no. 11 (2020): 4777-4787.
29. Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15, no. 1 (2014): 1929-1958.
30. Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In *International conference on machine learning*, pp. 448-456. PMLR, 2015.
31. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).