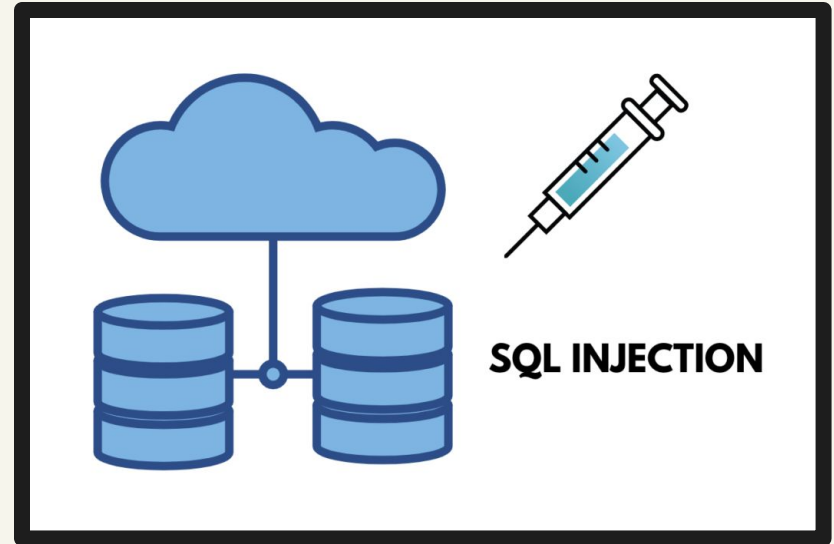


# Machine Learning Detection In SQL Injections

Armando Valdez

University of  
Minnesota Morris

April 13th 2025



# Outline of ML Detection in SQLi

## Introduction

- SQL and SQLi
- Stop SQLi
- Machine Learning
- Example of SQLi

## Model

- Training Neural Networks
- Improved Text-CNN
- Subsets of ML used in Improved Text-CNN
- Test & Results

## Conclusion-Takeaways

- Important to use machine learning to find SQLi
- Improvement in further model

## Questions

- References



# So what is SQL and SQL Injections?

Query- query is a command you send to a database to ask it to do something for example fetch data, insert new data, update existing data, or delete data.

SQL- Stands for Structured Query Language, it is a programming language that is used to talk to databases

SQL Injections(SQLi)- This is a hacking technique that uses harmful code to interfere with a databases.

# Why detection and defense are critical?

- In Open Worldwide Application Security Project (OWASP) says that SQLI are ranked #1 for over 20 years for databases to be hacked
- Many web applications use SQL to retrieve or store user data like login info, credit cards, private messages.
- Traditional rule-based methods can't keep up with evolving tools and hacking techniques.



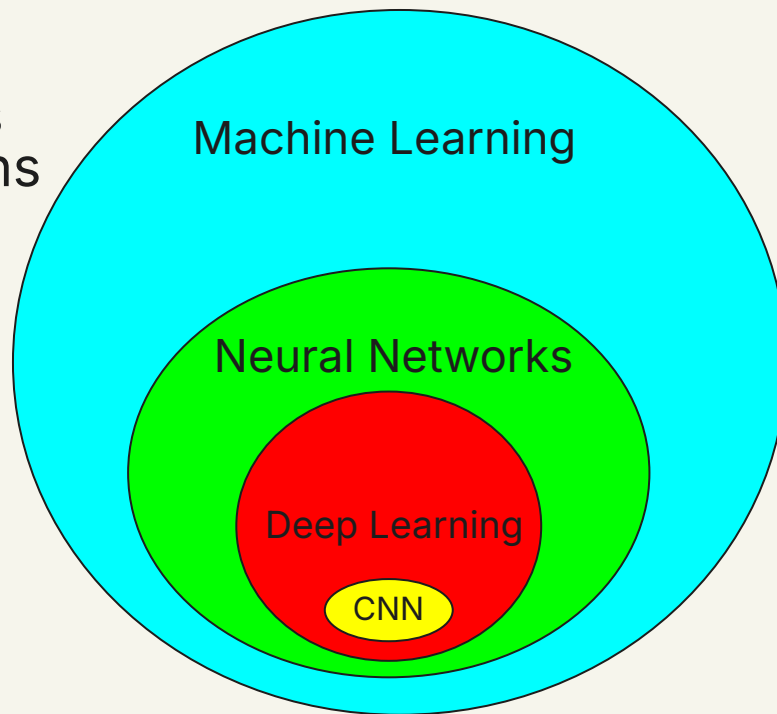
©2025 Gcore. All rights reserved.

# Introduction-Machine Learning

Machine Learning (ML)-is when algorithms learn patterns from data to make predictions or decisions without being explicitly programmed.

Deep Learning(DL)-is a subset of machine learning that uses multilayer neural networks to learn complex patterns from datasets.

Convolutional Neural Network(CNN)- is a deep learning model that finds patterns in data and is used for images, text, audio, and tasks like SQL injection detection.



# Understanding Boolean Logic (AND) & (OR)

TRUE AND TRUE = TRUE

FALSE AND TRUE = FALSE

FALSE AND FALSE = FALSE

TRUE OR TRUE = TRUE

FALSE OR TRUE = TRUE

FALSE OR FALSE = FALSE

-The AND operator in Boolean logic is used to combine two truth values.

-It returns True only if both values being compared are True.

-If either one or both values are False, the result of the AND operation will be False.

-The OR operator in Boolean logic is used to combine two truth values.

-It returns True if at least one of the values is True.

-The only time an OR operation results in False is when both values are False

# SQL Injection Example

- This checks if a user named Armando@gmail.com exists (TRUE) with that 12345 is the password (TRUE).

TRUE      SELECT \* FROM users  
WHERE username =  
Armando@gmail.com  
AND  
TRUE      password = 12345;

Login

Armando@gmail.com

12345

LOGIN

# SQL Injection Example

-So let's say the username is your email

-no idea what the password is?

-Adding OR 1 = 1 to the password

```
SELECT * FROM users
```

```
WHERE username =  
Armando@gmail.com
```

```
AND
```

```
password = 555 OR 1=1;
```

Login

Username

Password

LOGIN



# SQL Injection Example

- `SELECT * FROM users WHERE username = 'Armando@gmail.com' AND password = '555' OR '1'='1'`
- Coming back to Boolean
- `T AND (F OR T) → T AND T → T`

`SELECT * FROM users`

TRUE

`WHERE username = 'Armando@gmail.com'`

AND

FALSE OR TRUE

`password = '555' OR 1 = 1;`



# SQL Injection Example

```
SELECT COUNT(*) FROM wp_term_relationships,  
wp_posts WHERE wp_posts.ID =  
wp_term_relationships.object_id AND post_status IN  
( 'publish' ) AND post_type IN ( 'post ' ) AND  
term_taxonomy_id = 1 and 1=0 union select 1,  
concat_ws(0x3a,version() , user() , database() )  
,3,4,5,6,7,8,9,10,11,12--
```

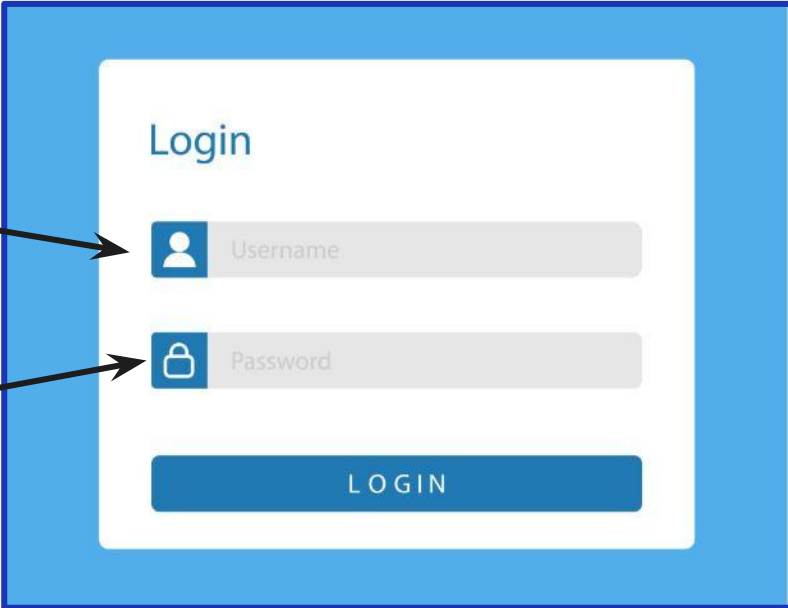


Diagram illustrating a login form interface. The form is titled "Login" and contains two input fields: "Username" (with a user icon) and "Password" (with a lock icon). Below the input fields is a "LOGIN" button. Arrows point from the SQL injection payload to the input fields, indicating the injection point.

# Training-Neural Networks

Layer – A step in a neural network that processes data, like extracting features or making predictions.

Used To Predict the

- Stock Market
- SQL injections

Loss Functions



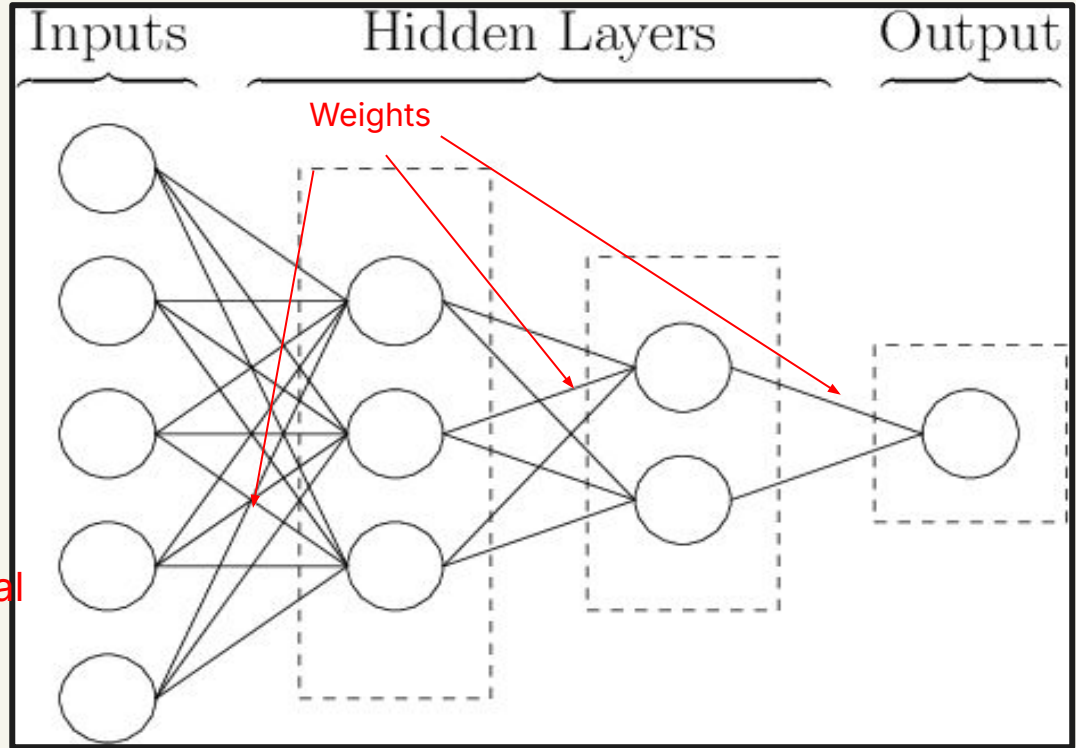
Back Propagation



Adjust weights from Hidden Layers neural outputs



Adjust weights from features



## The Improved Text-CNN Model-Subsets of Machine Learning

Improved Text-CNN- A deep learning model for text classification that uses convolution to extract features from text and includes an attention mechanism to focus on important patterns like SQL injection indicators.

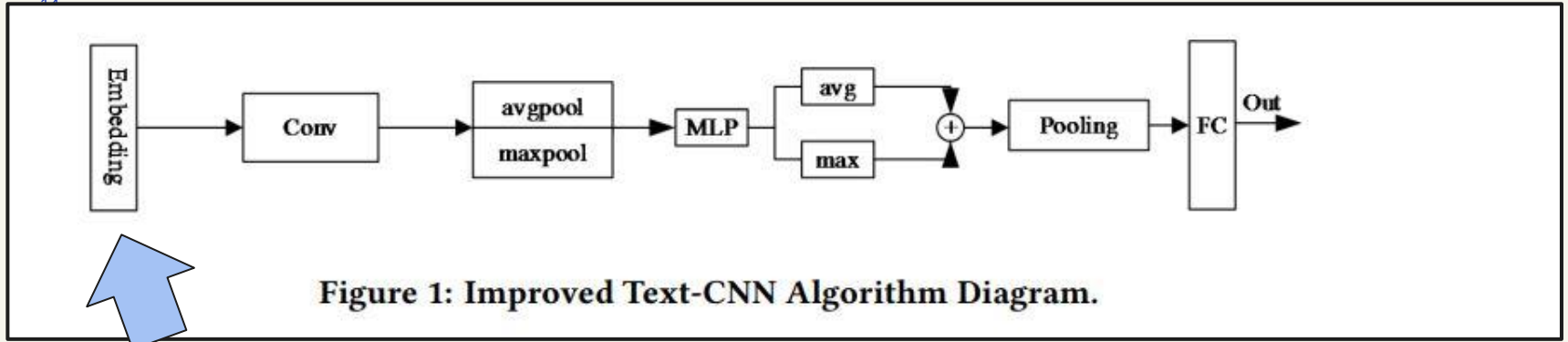
Feature- Information extracted from input text that helps the model decide if it's normal or a SQL injection attack.

Convolutional filters – Small pattern detector in a neural network that slide over the input text to identify suspicious phrases in SQL queries.

## The Improved Text-CNN Model-Subsets of Machine Learning

Kernel Size – The number of words a convolutional filter looks at once when scanning the input text.

Channel Attention Mechanism (CAM) – A method that helps the model focus on the most important feature maps before making a prediction.



© 2024 SQL Injection Attack Detection Based on Text-CNN

## Embedding-In Text-CNN

Converts words into numerical vectors so the model can understand and process them using Word2Vec.

Word2Vec – Converts each word or symbol into a vector of numbers.

These vectors are not random, similar words have similar vectors.

# Embedding

Neural networks like Improved Text-CNN can't read raw text so they need meaningful numbers that represent the text.

Word2Vec maps each token from a SQL query to a dense vector of numbers

## SQL query

```
SELECT * FROM users
WHERE password = 2
```

After tokenization

```
["SELECT", "*",  
"FROM", "users",  
"WHERE", "password",  
"=", "2"]
```

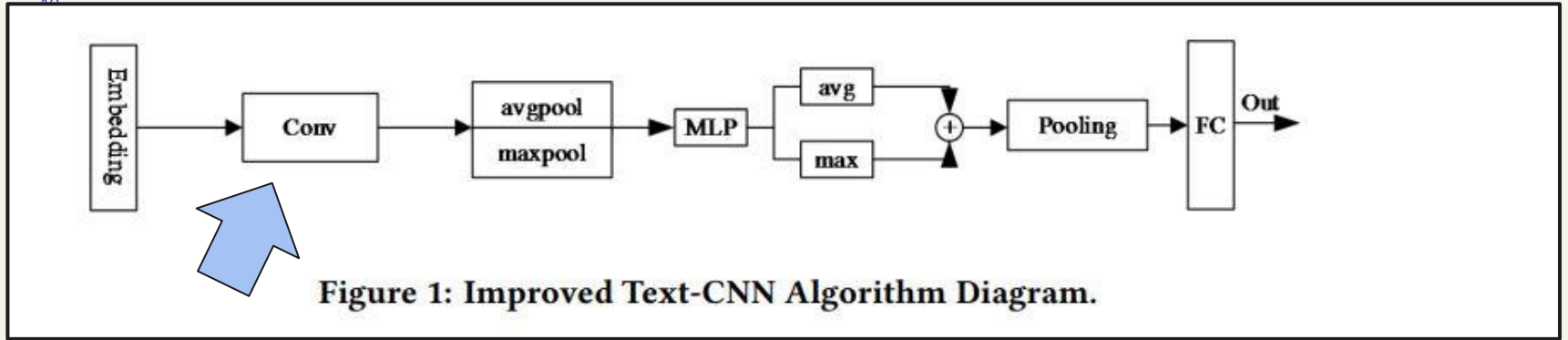
Each of these tokens gets converted into a Word2Vec embedding, like:

"SELECT" → [0.12, -0.45, 0.88, ..., 0.05]

"users"  $\rightarrow [0.31, 0.10, -0.74, \dots, -0.22]$

[illegible]

© 2008-2025 ResearchGate GmbH. All rights reserved.



## Conv (Convolutional Layers)- In Text-CNN

-Applies multiple filters of varying sizes in order to capture local features like tricky phrases or word patterns from the embedded text.

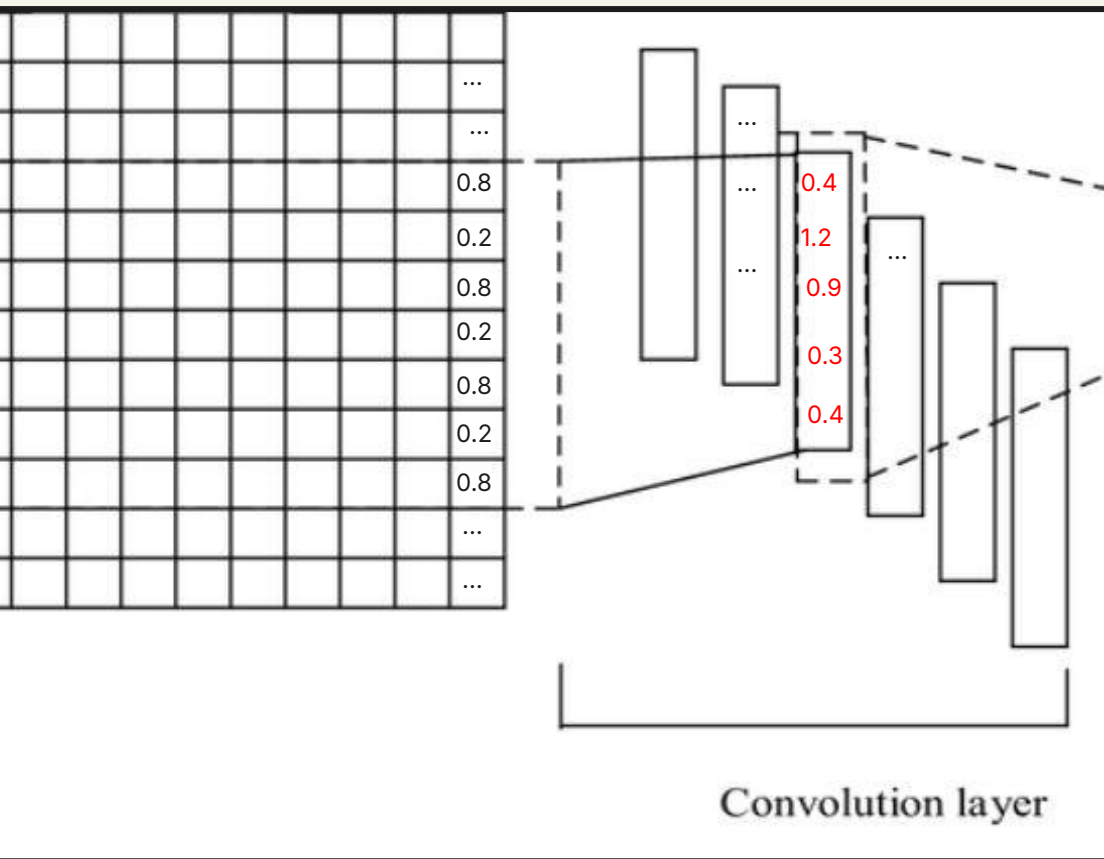
-Each filter produces a feature map.

Low-level local features- These are small patterns that the model picks up

High-level global features- These come from combining multiple local features into a more meaningful structure.



## Conv (Convolutional Layers)



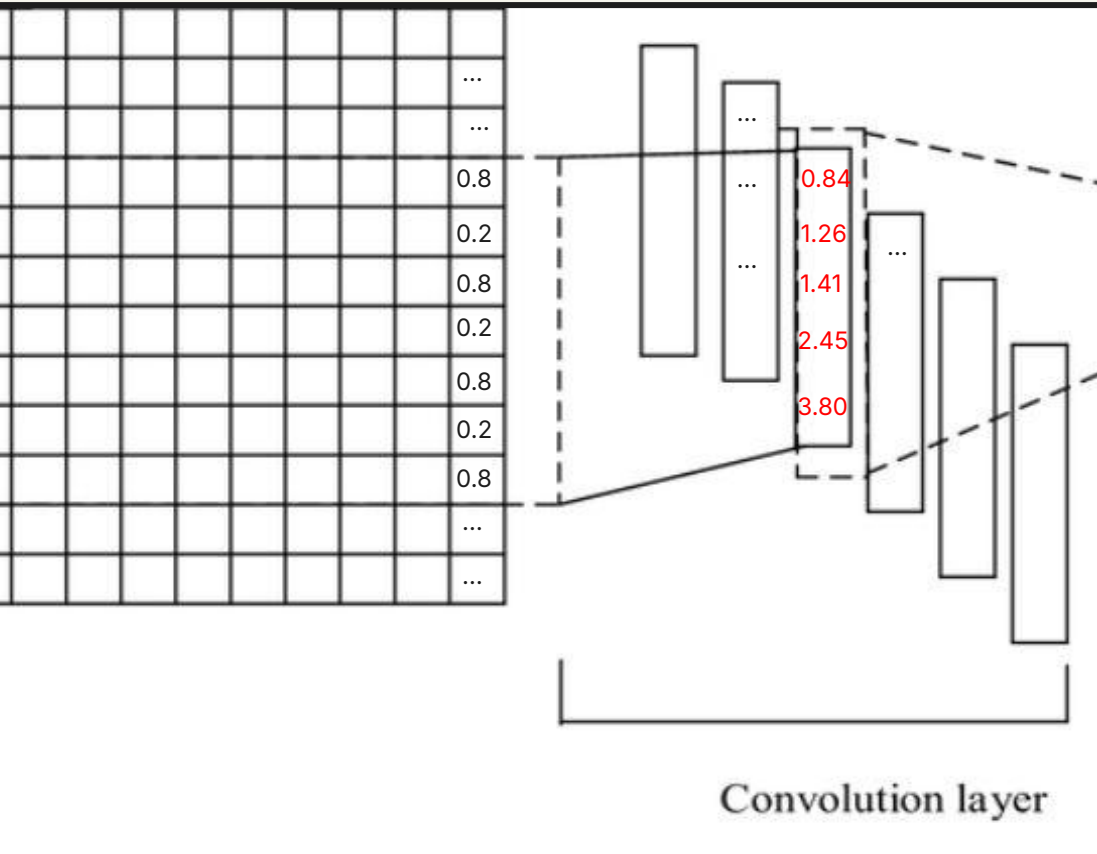
The Convolution Layer starts filtering -detect local patterns

Let's say that the token (numbers) has a pattern

[0.84, 1.26, 1.41, 2.45, 3.80,....]

A potential High-level global feature

## Conv (Convolutional Layers)



Kernel size = 3,

(SELECT \* FROM) → 0.84

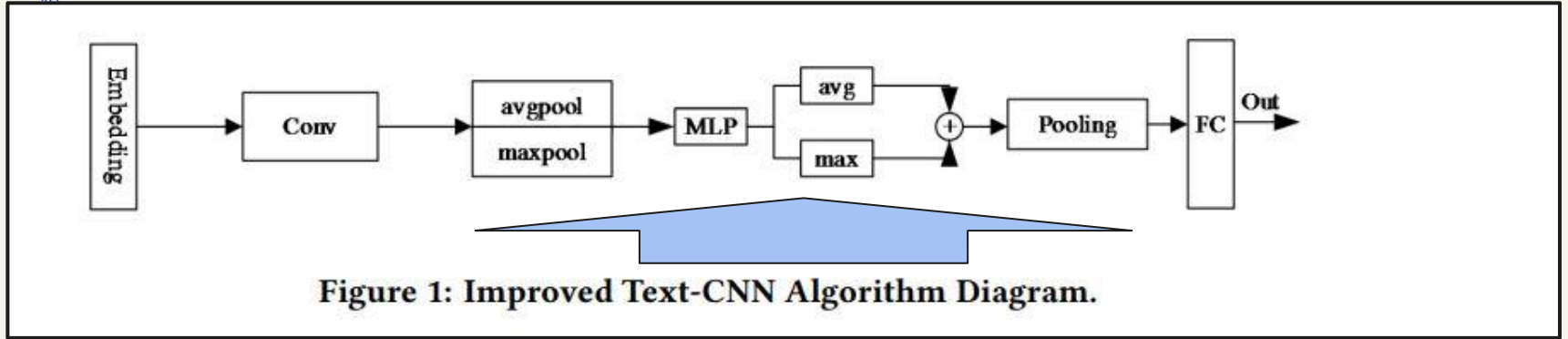
(\* FROM password) → 1.26

(FROM password = ) → 1.41

(password=2 OR) → 2.45

(OR 1=1) → \*\*3.80\*\*

This is the feature map:  
[0.84, 1.26, 1.41, 2.45, 3.80]



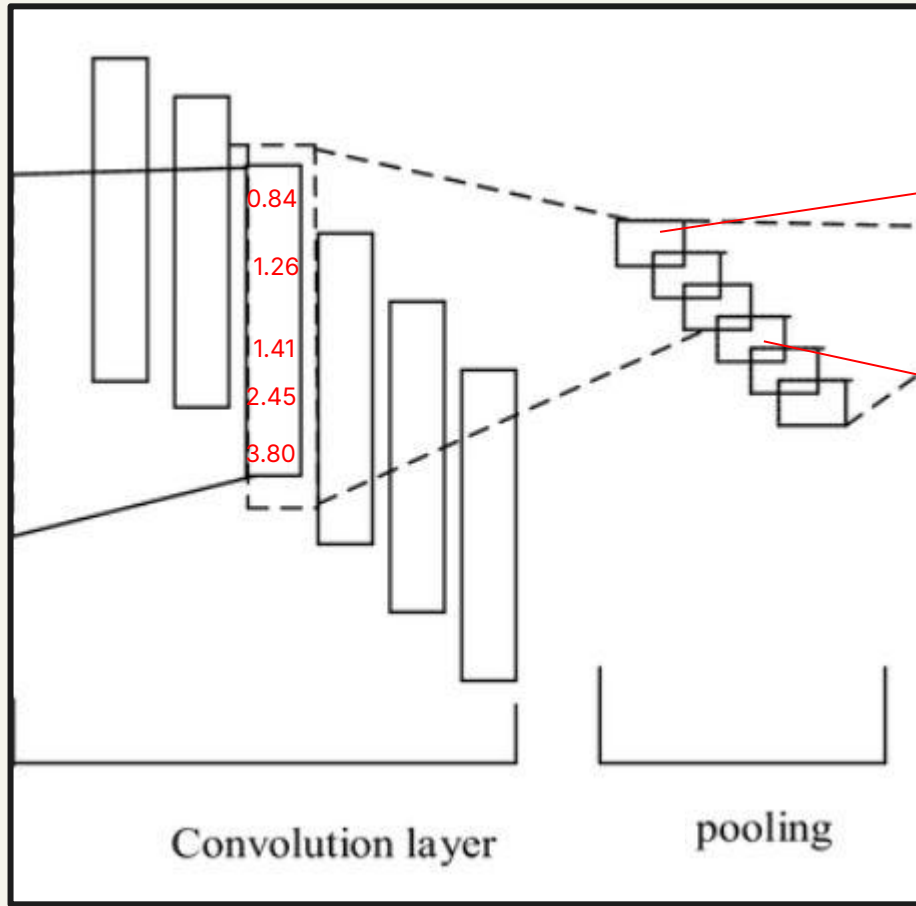
## Pooling Layers

Pooling is a process in neural networks that reduces the size of the data while keeping the most important information.

Max pooling- picks the largest value in a section of the data.

Average pooling- takes the average (mean) of values in each section.

Multi-Layer Perceptron- neural network made of fully connected layers that learns to combine and weight features(important scores).



Max pooling- picks the largest value in a section of the data.

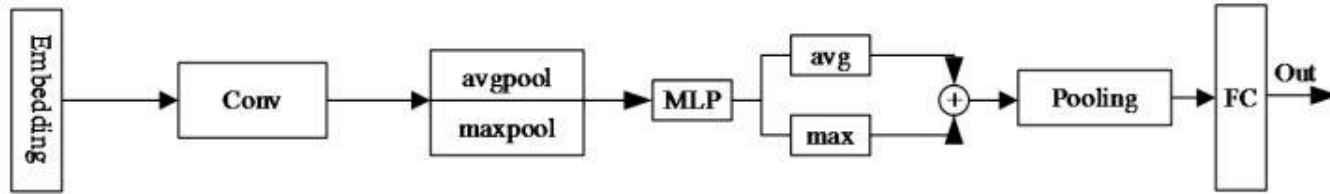
$[0.84, 1.26, 1.41, 2.45, 3.80]$

Takes the max number:

Max pool  $\rightarrow \max([0.84, 1.26, 1.41, 2.45, 3.80]) = 3.80$

Average Pooling- takes the average (mean) Value

$[0.84, 1.26, 1.41, 2.45, 3.80] \rightarrow 0.84 + 1.26 + \dots = 9.76 \rightarrow 9.76/5 = 1.952$



**Figure 1: Improved Text-CNN Algorithm Diagram.**

### FC (Fully Connected Layer)

Takes the final feature vector (after convolution, attention, and pooling) and makes the final decision about what class the input belongs to with CAM & Softmax.

Softmax Layer – Converts raw scores into probabilities for each class using math, helping the model make a final prediction.

## FC (Fully Connected Layer) + CAM Layer

Max pooling  $\rightarrow$  [3.80]

Average pooling  $\rightarrow$  [1.952]

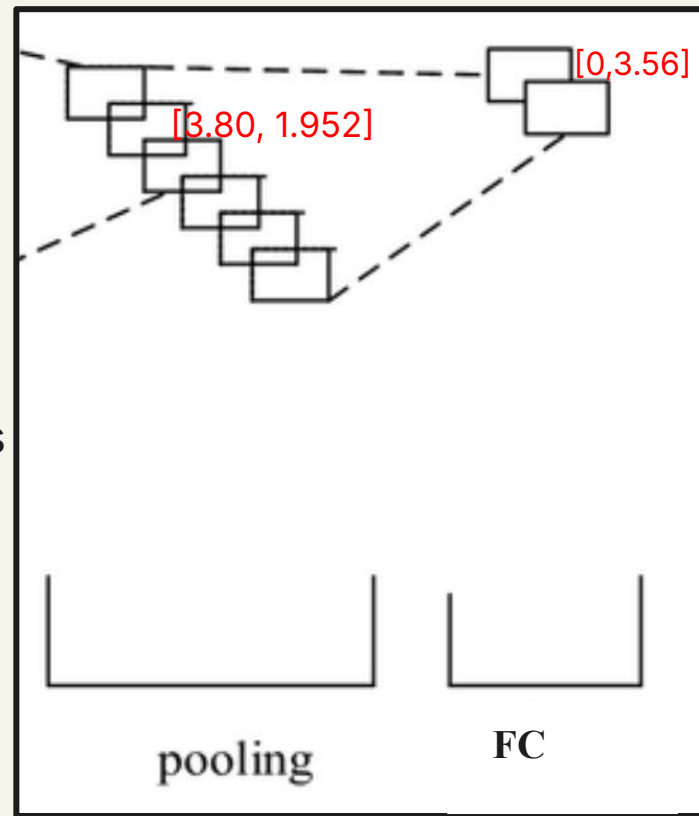
- These values are combined and fed into a small neural network
- That small neural network learns its own **weights** and **bias** to compute attention weights
- These attention weights are used to scale the feature maps

Weight = [1.2], Bias = [-1.0]

Linear output =  $3.80 \times 1.2 + (-1.0) = [3.56]$

Weight = [0], Bias = [0]

Linear output =  $3.80 \times 0 + (0) = [0]$



## FC + SoftMax Formula

Formula is used to measure probability, these correspond to:

Class 0 = Less likely

Class 1 = More likely

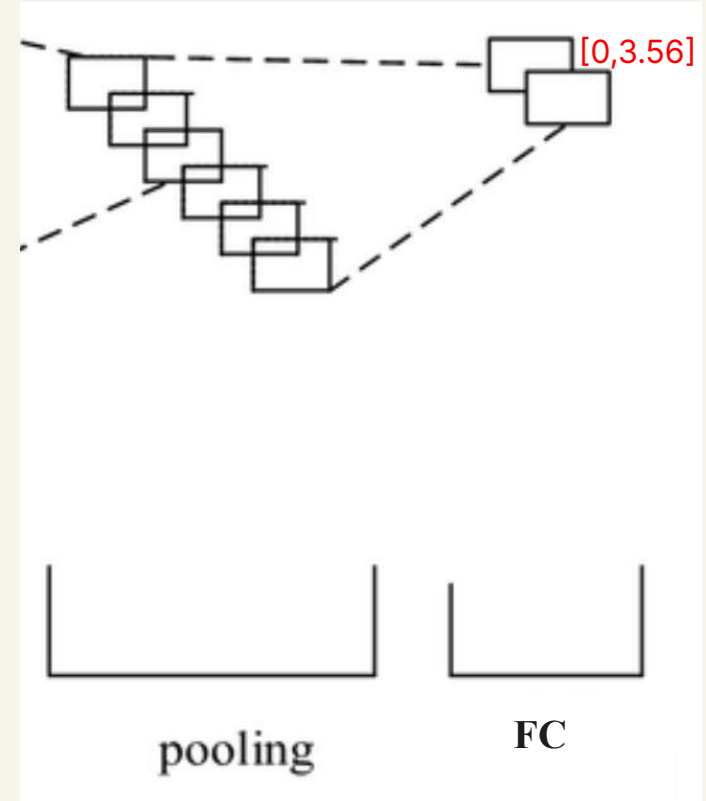
$$\text{Softmax}(x) = \frac{e^x}{1 + e^{\text{maxpool}}}$$

$$\text{Softmax}(0) = \frac{e^0}{1 + e^{3.56}} \rightarrow \frac{1}{1 + e^{3.56}} \rightarrow \frac{1}{36.2} \rightarrow 0.0276$$

$$\text{Softmax}(3.56) = \frac{e^{3.56}}{1 + e^{3.56}} \rightarrow \frac{35.2}{36.2} \rightarrow 0.9724$$

Output-Probability & Result

[0.0276 → Normal, 0.9724 → SQL Injection]



## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

Real world labeled dataset called Libinjection project from Github and in this dataset it contains both Normal SQL queries and SQL Injection attacks

### Training Set

- This is the data the model actually learns from.
- During training the model

### Validation Set

- Only used to monitor performance during training.
- "Is the model learning the right things?"

### Testing Set

- This data is completely unseen during training and validation.



## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### Models to Compare

- Text-RNN- Baseline using recurrent neural networks, processes word-by-word, remembers order
- CNN-Standard convolutional model, simple pattern matcher, no embeddings
- Text-CNN- Traditional Text-CNN model, Word2Vec + filters
- Improved Text-CNN- Text-CNN + Attention (CAM)

## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### They evaluated performance using common classification metrics

Positive- Malicious Code

Negative- Normal Code

Machine learning \ Manual counting	True	False
True	True Positive (TP)	False Positive (FP)
False	False Negative (FN)	True Negative (TN)

© 2008-2025 ResearchGate GmbH. All rights reserved.

## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### They evaluated performance using common classification metrics

#### **Accuracy**

- How many did the model get right?
- The fewest number of mistakes

$$Accuracy = \frac{\text{Total correct guesses}}{\text{Total guesses}} = \frac{TP + TN}{TP + TN + FP + FN}$$

## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### They evaluated performance using common classification metrics

#### **Precision**

- Measures how many of the predicted positive cases are actually positive.
- How precise you got positive right.

$$Precision = \frac{\text{Correct positive guesses}}{\text{Total positive guesses}} = \frac{TP}{TP + FP}$$

## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### They evaluated performance using common classification metrics

#### **Recall**

- Measures how many actual positive cases the model correctly identifies.
- How good are the positives recalled.

$$Recall = \frac{\text{Correct positive guesses}}{\text{All positive labels}} = \frac{TP}{TP + FN}$$

## Testing- Improved Text-CNN

**Table 1: Data set Classification.**

Data Set	Normal Request	SQL Injection	Total
Training Set	25500	20500	46000
Validation Set	11000	11000	22000
Test Set	5000	5000	10000

### They evaluated performance using common classification metrics

#### **F1**

- The harmonic mean of precision and recall, balancing both metrics.
- Best for imbalanced datasets where accuracy is misleading.

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

## Results- Improved Text-CNN

**Table 2: Experimental Results Table.**

Algorithm	Precision	Accuracy	Recall	F1
Text-RNN	0.9079	0.9078	0.9076	0.9077
CNN	0.9113	0.9111	0.9101	0.9108
Text-CNN	0.9123	0.9121	0.9110	0.9116
Improved Text-CNN	0.9245	0.9240	0.9229	0.9237

**Across the board, Improved Text-CNN had the highest score**

### Improved Text-CNN

Precision = 92.40%

Accuracy = 92.45%

Recall = 92.29%

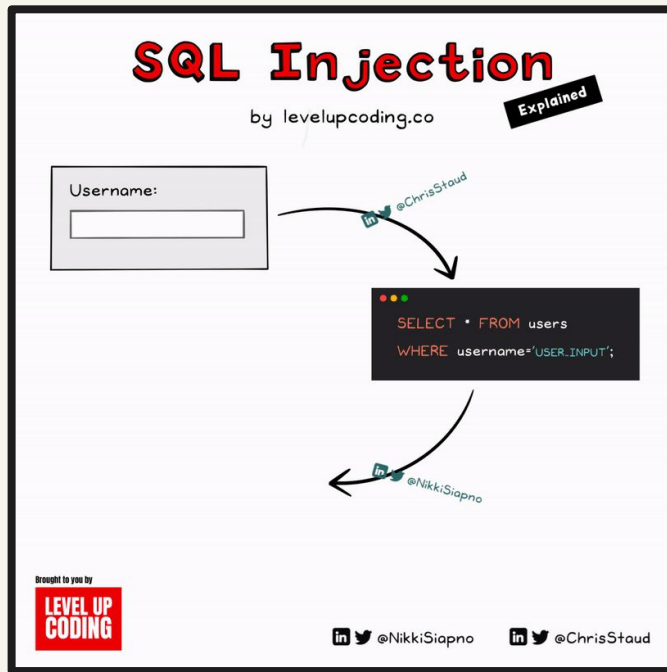
F1 = 92.37%

## Takeaways- Improved Text-CNN

Traditional rule-based methods can't keep up with evolving attacks.

Machine learning models especially deep learning are important because:

- They can learn from data rather than relying on hard-coded rules.
- They provide better generalization across different types of queries and user inputs.
- Moves a lot faster than humans and traditional rules.
- Make tools that actually make SQLi in order to find them before they attack.



©Levelupcoding.co 2020-2025



# Questions?

## References

Wei Zhao, Junling You, and Qinghui Chen. 2024. SQL Injection Attack Detection Based on Text-CNN. <https://doi.org/10.1145/3665348.3665398>

Rajesh Sharma and Mia Tang. 2024. Machine Learning & Neural Networks. <https://doi.org/10.1145/3664475.3664574>

Risto Miikkulainen. 2024. Evolution of Neural Networks. <https://doi.org/10.1145/3638530.3648407>

John Irungu, Steffi Graham, Anteneh Girma, and Thabet Kacem. 2023. Artificial Intelligence Techniques for SQL Injection Attack Detection. <https://doi.org/10.1145/3583133.3590531>

Andrea Ferigo and Giovanni Iacca. 2023. Self-Building Neural Networks. <https://doi.org/10.1145/3583133.3590531>

Sharath S Hebbar 2017 Softmax for Intermediate CNN Layers. <https://medium.com/@sharathhebbbar24/softmax-for-intermediate-cnn-layers-f6d3b8b7d1d2>

[https://www.researchgate.net/figure/Character-level-embedding-Conv-char-Emb-with-CNN\\_fig2\\_331836693](https://www.researchgate.net/figure/Character-level-embedding-Conv-char-Emb-with-CNN_fig2_331836693)