

BAB II

Tinjauan Pustaka dan Dasar Teori

2.2.1. Kajian Hasil Penelitian Terdahulu

Penelitian ini mengawali dengan membahas hasil akurasi dari model deteksi *packet format* dan *WLAN modulation classification* yang dikembangkan menggunakan pendekatan *Convolutional Neural Network* (CNN). Model ini secara spesifik belajar untuk mengenali *packet format* WLAN seperti *Legacy*, *High Throughput* (HT), *Very High Throughput* (VHT), *High Efficiency* (HE), serta skema modulasi seperti DL, FRC, FR1, dan QPSK untuk WLAN, dan DL, FRC, FR1, dan 64QAM untuk 5G. Dalam tahap pengujian yang dilakukan terhadap model-model yang telah dirancang, hasilnya menunjukkan bahwa untuk deteksi *packet format* WLAN, model mencapai tingkat akurasi sebesar 99%, sementara untuk *WLAN modulation classification*, model ini berhasil mencapai tingkat akurasi sebesar 90% (Putra dkk, 2021).

Penelitian kedua membahas tentang pembuatan model dengan pendekatan *Multi-Task Learning* menggunakan *Convolutional Neural Network* untuk *Packet Collision Avoidance* pada 802.11 WLAN dengan mengaplikasikan *Packet Format Detection* dan *Signal Modulation Classification*. Model ini mempelajari *packet format* dan modulasi yang terdapat pada WLAN. Pada *packet format* dan modulasi tersebut ditambahkan ciri yaitu SNR (*Signal Noise Ratio*) dari 4dB sampai 24dB. Hasil dari penelitian ini adalah mendapatkan akurasi sebesar 98.93% untuk *Packet Format Detection* dan mendapatkan akurasi sebesar 88.53% untuk *Signal Modulation Classification* (Putra dkk, 2022).

Penelitian selanjutnya membahas tentang hasil akurasi dari klasifikasi *wireless signal* menggunakan metode *Long-Short Term Memory* (LSTM). Model ini belajar dari amplitudo domain waktu dan informasi fase dari skema modulasi yang ada dalam data pelatihan tanpa memerlukan fitur ahli seperti momen siklik orde tinggi. Analisis menunjukkan bahwa model yang diusulkan menghasilkan akurasi klasifikasi rata-rata mendekati 90% pada berbagai kondisi rasio *noise* mulai dari 0 dB hingga 20 dB (Ranjendran dkk., 2018).

Penelitian selanjutnya membahas penggunaan *deep learning* untuk mendeteksi multi-sinyal dan mengklasifikasi modulasinya. Pada penelitian ini dibuat 2 jenis model yaitu jaringan *Single Shot Multibox Detector* (SSD) untuk mendeteksi sinyal dan *Multi-inputs Convolutional Neural Networks* (CNNs) untuk klasifikasi modulasi sinyal tersebut. Hasil terbaik yang didapatkan pada model pertama yaitu *Single Shot Multibox Detector* (SSD) untuk mendeteksi sinyal adalah dengan *input IoU* sebesar 0.5 yang mendapatkan akurasi 60% pada 0dB dan 96% pada 10dB. Lalu akurasi terbaik yang didapatkan pada model kedua yaitu *Multi-inputs Convolutional Neural Networks* (CNNs) untuk klasifikasi modulasi sinyal adalah 80% pada 0dB dan 100% pada 10dB (Zha dkk., 2019).

Penelitian selanjutnya membahas mengenai pengembangan pengklasifikasian modulasi sinyal dengan menggunakan tiga arsitektur yaitu, *Convolutional Long Short-term Deep Neural Network* (CLDNN), *Long Short-Term Memory neural network* (LSTM), dan *Deep Residual Network* (ResNet). Penelitian ini menghasilkan nilai akurasi klasifikasi tipikal sekitar 90% pada SNR tinggi (Ramjee dkk., 2019).

Tabel 2.1 Arsitektur yang digunakan pada penelitian terdahulu

Penulis	Arsitektur yang digunakan	Akurasi hasil penelitian
Putra dkk, 2021	CNN	<i>Packet format detection</i> 99%, <i>WLAN modulation classification</i> 90%.
Putra dkk, 2022	CNN	<i>Packet format detection</i> 98,93%, <i>Signal modulation classification</i> 88.53%.
Rajendran dkk, 2018	LSTM	Model dengan LSTM yang memiliki hasil rata-rata mendekati 90%.
Zha dkk, 2019	<i>Single Shot Multibox Detector</i> (SSD) dan <i>Multi-input CNN</i>	SSD sebesar 60% pada 0dB, 96% pada 10dB. Multi-input CNN sebesar 80% pada 0dB, 100% pada 10dB.

Ramjee dkk, 2019	<i>Convolutional Long Short-term Deep Neural Network</i> (CLDNN), LSTM dan ResNet	Pada ketiga arsitektur mendapatkan rata-rata 90% pada SNR tinggi
------------------	---	--

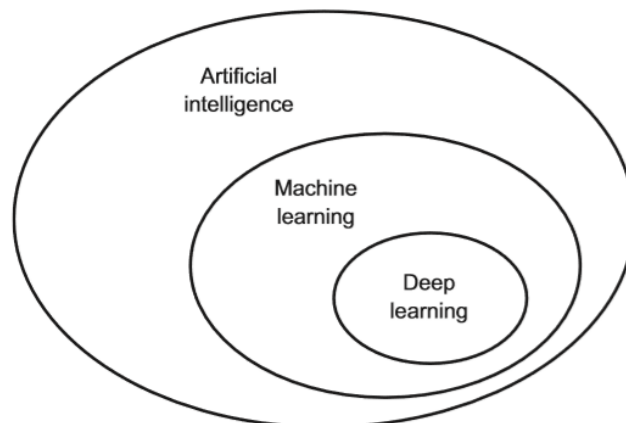
2.2 Artificial Intelligence

Artificial Intelligence (AI) merupakan bagian dari ilmu yang dipelajari di bidang komputer dimana, dengan adanya AI mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. AI bertujuan untuk mengetahui atau memodelkan proses berpikir manusia dan mendesain mesin sehingga bisa menirukan perilaku manusia. *Artificial Intelligence* merupakan ilmu yang sangat luas, didalamnya terdapat bidang-bidang lain seperti Sistem Pakar, Natural Language Processing, *Speech Recognition*, *Computer Vision*, *Machine Learning* (ML), dan *Deep Learning* (DL). Dalam AI, secara umum terdapat dua bagian utama, yaitu (Sammut dan Webb, 2011):

- *Knowledge Base* (Basis Pengetahuan) yaitu berisi tentang fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.
- *Inference Engine* (Motor Inferensi) yaitu kemampuan menarik kesimpulan berdasarkan pengalaman atau latihan yang sudah dilakukan.

2.2.1 Deep Learning

Deep learning merupakan salah satu bagian dari kecerdasan buatan dan *machine learning*. Perkembangan konsep *deep learning* timbul sebagai respons terhadap keterbatasan yang ditemukan dalam paradigma *machine learning*. *Deep learning* sendiri merupakan sistem yang dibangun dengan algoritma yang terinspirasi dari cara kerja struktur biologis syaraf manusia untuk menyelesaikan suatu permasalahan. *Deep learning* dapat diimplementasikan untuk pengolahan data *text*, data gambar, data suara, atau data *video*. Hubungan antara *artificial intelligence*, *machine learning*, dan *deep learning* dijelaskan diagram venn pada Gambar 2.1 berikut.



Gambar 2.1 *Artificial intelligence, machine learning, dan deep learning*
(Lu and Zheng, 2017)

Machine learning dan *deep learning* menggunakan prinsip kerja *artificial intelligence*. *Deep learning* menggunakan prinsip kerja dari *machine learning*. Disini, perbedaan dari *deep learning* dan *machine learning* berada pada cara kerjanya. Pada *deep learning* menggunakan banyak *hidden layer* untuk mengolah data sesuai dengan kaidah jaringan syaraf manusia. Sedangkan pada *machine learning* jumlah *hidden layer* lebih sedikit daripada *hidden layer* pada *deep learning* (Goodfellow dkk, 2016).

Selain pada jumlah *hidden layer*, perbedaan antara *machine learning* dan *deep learning* terletak pada pendekatan sistem untuk memperoleh kecerdasan. Pada *Machine learning*, kecerdasan sistem diperoleh melalui pemrograman logis, penalaran induktif berdasarkan aturan dasar, ataupun dari sifat data masukan. Sedangkan pada *deep learning* sistem mendapatkan kecerdasan dari pengolahan dan pengenalan data pada tahap *hidden layer* (Mathew dkk, 2021).

2.2.2. ***Supervised Learning***

Supervised learning merupakan salah satu pendekatan terhadap data yang akan dilatih menggunakan *deep learning*, dan untuk pendekatan lainnya yaitu *unsupervised learning*, dan *semi-supervised learning*. Perbedaan yang paling mencolok antara ketiga pendekatan tersebut adalah pada bagian *labeling* data. Pada *supervised learning*, data yang akan dilatih diberikan juga label, keterangan atau pola data terkait. Pada *unsupervised learning*, data yang akan dilatih tidak diberi label, sehingga sistem yang dibuat akan mengelompokkan data berdasarkan

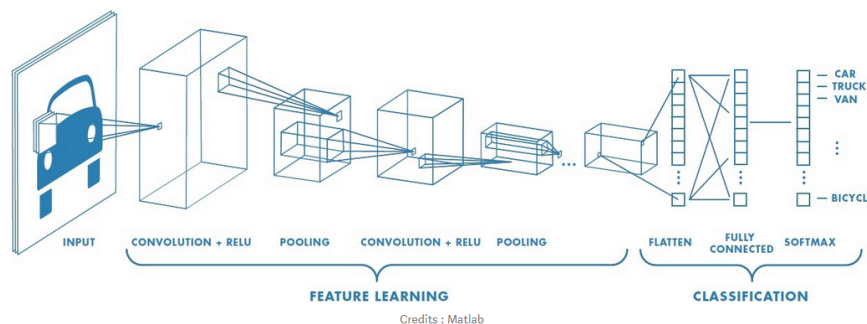
kemiripan data. Sedangkan untuk *semi-supervised learning* merupakan percampuran antara keduanya, yang artinya data yang akan dilatih sebagian diberikan label dan sebagian tidak diberikan label.

Pendekatan yang sering digunakan khususnya dalam *deep learning* adalah *supervised learning*. Label pada setiap data akan mempermudah sistem untuk mengenali pola dari data yang dimasukkan. Pada *supervised learning*, data yang akan dilatih telah mengenali target dari setiap data. *Supervised learning* dapat digunakan untuk *deep learning* dan *machine learning* antara lain: klasifikasi, regresi, *sequence generation*, *syntax tree prediction*, deteksi objek, atau segmentasi gambar (Vasilev dkk., 2019).

2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur dari *deep learning*. Arsitektur CNN biasanya digunakan untuk pemrosesan data dalam jumlah besar. Seperti halnya pada *neural network* yang lain, arsitektur ini juga memiliki proses *fully-connected layer* atau *hidden layer*. Semakin banyak *layer* dalam *hidden layer* akan berpengaruh pada jumlah parameter dari model.

Secara umum, *Convolutional Neural Network* (CNN) merupakan bentuk pengembangan dari *Multi Layer Perceptron* (MLP) yang di desain untuk pengolahan data *text* dengan jumlah banyak. Dalam CNN, data akan diolah terlebih dahulu dalam proses konvolusional dahulu sebelum masuk ke proses klasifikasi menggunakan *fully-connected layer*. Terdapat 2 proses utama yaitu proses *feature learning* dan *classification* seperti yang ditampilkan pada Gambar 2.2.



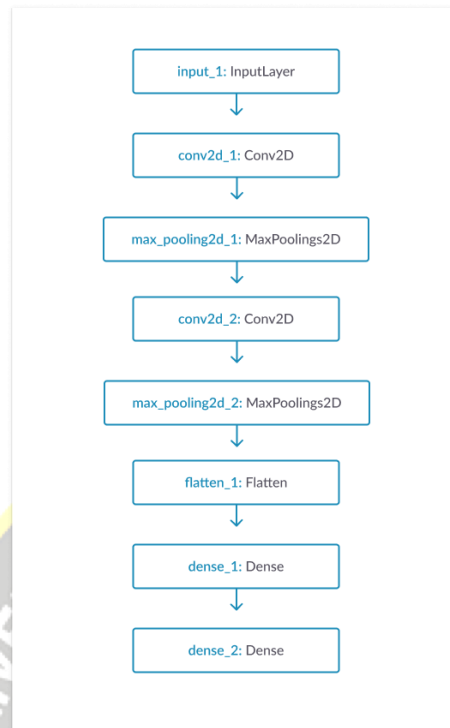
Gambar 2.2 Proses CNN (Li dkk, 2021)

Berdasarkan gambaran prosesnya, Convolutional Neural Network (CNN) dapat terbagi menjadi dua tahap utama, yaitu *feature learning* dan *classification*. Proses *feature learning* lebih berfokus ke pengolahan data masukan, sedangkan proses *classification* lebih berfokus ke proses pengelompokan data sesuai dengan keluaran yang diinginkan. Dalam proses *feature learning* terdapat proses konvolusional dengan nilai aktivasi dan proses *pooling*. Hasil akhir dari proses *feature learning* akan diolah menjadi nilai *flatten* sebelum masuk ke proses klasifikasi atau *fully-connected layer*. Proses *classification* terdiri dari banyak *hidden layer* yang menggunakan konsep *neural network*. Setiap *layer* akan terhubung ke *layer* berikutnya sampai ke *layer* terakhir (Li dkk, 2021).

2.3.1 Convolutional Neural Network Sequential

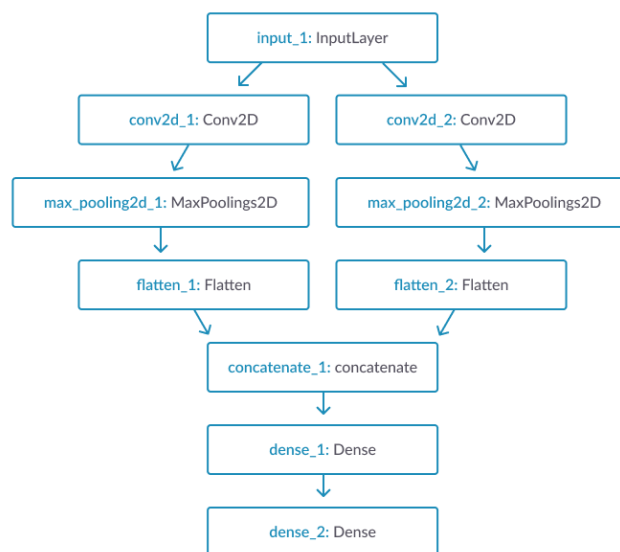
CNN *Sequential* merupakan salah satu pemodelan dalam arsitektur *Convolutional Neural Network* (CNN) seperti halnya pada model lainnya, antara lain AlexNet, VGGNet, ResNet, Inception V3, MobileNet. Penggunaan model eksternal bertujuan untuk mempermudah proses pemodelan CNN. Dengan memasukkan nilai bobot dari AlexNet, VGGNet atau model yang lain akan membuat model akan lebih banyak parameter latih karena memasukkan nilai bobot yang sudah diuji sebelumnya. Jumlah *layer* akan menjadi bertambah banyak, tetapi kenaikan waktu latih yang tidak signifikan.

Menggunakan model eksternal akan berpengaruh pada proses *feature learning* saja, untuk proses *classification* tetap dibuat dengan jumlah *layer* yang sesuai dengan kebutuhan. Disisi lain, pembuatan model dengan menggunakan CNN *Sequential* tanpa menggunakan model eksternal diharuskan membangun semua susunan *layer* secara *manual* dari *input*, *convolutional*, *polling*, *flatten*, hingga *output*. Kelebihan dari model CNN *Sequential* yaitu sebagai peneliti dapat bebas menentukan jumlah *layer* sesuai kebutuhan. Kekurangan dari CNN *Sequential* yaitu dalam proses latih akan membutuhkan waktu yang sebanding dengan jumlah *layer* dan parameternya. Terdapat beberapa bentuk *graph* yang dapat dimodelkan menggunakan CNN *Sequential* antara lain *linear layer*, *shared input layer* dan *multiple input layer*.



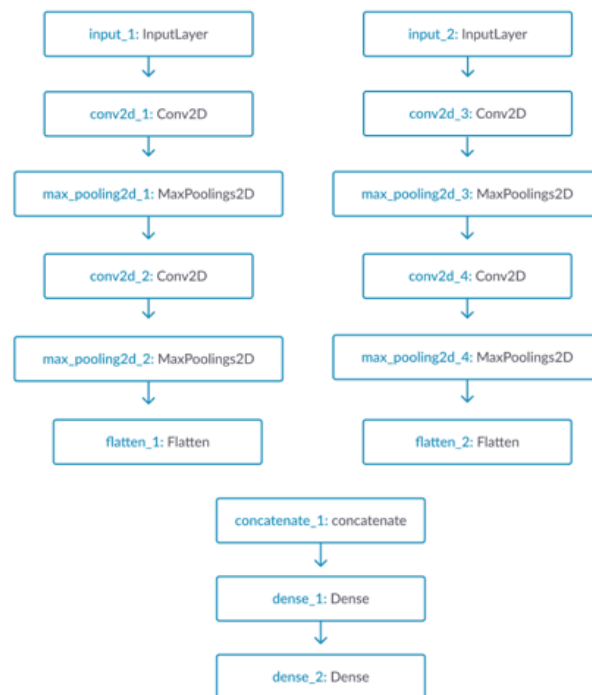
Gambar 2.3 *Linear layer* (Sokolova and Lapalme, 2009)

Seperti yang ditampilkan pada Gambar 2.3 tentang *linear layer*, setiap tahap dari *input* sampai *output* hanya dilakukan satu kali secara *linear*. Nilai *flatten* dari proses konvolusional akan langsung diproses menuju proses *fully-connected layer* tanpa ada proses *concatenation*.



Gambar 2.4 *Shared input layer* (Sokolova and Lapalme, 2009)

Seperti yang ditampilkan pada Gambar 2.4 tentang *shared input layer*, nilai *input* akan diolah dalam proses konvolusional dua kali dan akan menghasilkan 2 nilai *flatten*. Sebelum masuk ke tahap *fully-connected layer*, kedua nilai *flatten* akan melalui proses *concatenation* dengan tujuan mendapatkan satu nilai *flatten* akhir yang akan menjadi masukan dari tahap selanjutnya.



Gambar 2.5 *Multiple input layer* (Sokolova and Lapalme, 2009)

Seperti yang ditampilkan pada Gambar 2.5 tentang *multiple input layer* terdapat dua *layer* masukan yang masing masing masukan akan diproses dalam konvolusional sampai nilai *flatten* masing-masing. Seperti halnya pada *shared input layer*, nilai *flatten* yang dihasilkan berjumlah 2. Sebelum masuk ke proses *fully-connected layer* kedua nilai tersebut akan diolah dalam tahap *concatenation* agar mendapatkan nilai *flatten* akhir (Sokolova dan Lapalme, 2009).

2.3.2 Confusion Matrix

Confusion Matrix merupakan metode yang digunakan untuk menentukan nilai akurasi dari hasil klasifikasi. Dalam *confusion matrix* terdapat beberapa istilah seperti yang dituliskan pada Tabel 2.2 yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* (TP)

yaitu data dimana antara prediksi dengan kenyataan sesuai. *True Negative* (TN) yaitu data diluar kategori uji yang memiliki kecocokan antara prediksi dan kenyataan. *False Positive* (FP) yaitu data diluar kategori uji yang diprediksi sesuai dengan kategori uji. Sedangkan *False Negative* (FN) yaitu data kategori uji yang diprediksi diluar kategori uji.

Tabel 2.2 Keluaran klasifikasi (Haghighi dkk, 2018)

<i>Data class</i>	<i>Classified as pos</i>	<i>Classified as neg</i>
<i>Pos</i>	<i>true positive (tp)</i>	<i>false negative(fn)</i>
<i>Neg</i>	<i>False positive (fp)</i>	<i>true negative (tn)</i>

Dari keempat istilah pada Tabel 2.2, dapat dilakukan operasi perhitungan untuk mencari nilai *precision*, nilai *recall*, dan nilai *accuracy*. Ketiga nilai tersebut sebagai analisis uji dari klasifikasi yang dilakukan. Nilai *precision* merupakan perbandingan antara nilai TP dengan jumlah data yang diprediksi kategori uji (2.1). Nilai *Recall* yaitu perbandingan nilai TP dengan jumlah data kategori uji (2.2). Sedangkan nilai *accuracy* merupakan perbandingan total dari semua data klasifikasi dengan aturan yang sudah ditentukan (2.3) (Haghighi dkk, 2018).

$$Precision = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l (FP_i + TP_i)} \quad (2.1)$$

$$Recall = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l (TP_i + FN_i)} \quad (2.2)$$

$$Acuracy = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{l} \quad (2.3)$$

Keterangan:

TP = Jumlah data true positif

TN = Jumlah data true negatif

FP = Jumlah data false positif

FN = Jumlah data false negatif

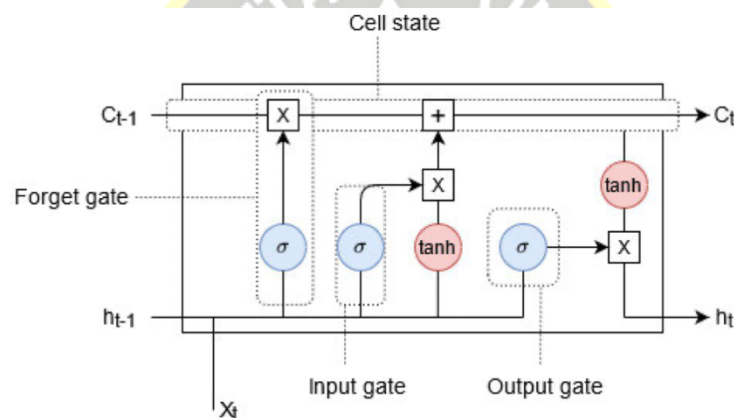
i = Urutan iterasi data

l = Jumlah kategori data

2.4 Long Short-Term Memory (LSTM)

Arsitektur LSTM (*Long Short-Term Memory*) adalah jenis arsitektur *Recurrent Neural Network* (RNN) yang dirancang khusus untuk mengatasi masalah perhitungan yang melibatkan urutan data atau data berurutan. LSTM telah terbukti sangat efektif dalam memproses dan memahami urutan data yang kompleks, seperti teks, suara, dan sinyal waktu nyata (Subray dkk., 2021).

Arsitektur LSTM terdiri dari satu atau beberapa unit LSTM, yang setiap unitnya memiliki struktur internal yang kompleks. Seperti yang tertera pada Gambar 2.6, setiap unit LSTM memiliki tiga komponen utama: *forget gate* (gerbang lupakan), *input gate* (gerbang masukan), dan *output gate* (gerbang keluaran).



Gambar 2.6 LSTM Cell (Subray dkk., 2021)

Forget Gate: Gerbang lupakan bertanggung jawab untuk memutuskan seberapa banyak informasi yang harus dilupakan oleh *unit* LSTM. Gerbang ini mengambil *input* berupa *input* dari *unit* sebelumnya (*input* sekuensial) dan *output* dari *unit* LSTM sendiri pada langkah waktu sebelumnya. *Output* gerbang lupakan adalah bilangan antara 0 dan 1 yang mengindikasikan seberapa banyak informasi yang akan diabaikan dalam pemrosesan saat ini.

Input Gate: Gerbang masukan mengontrol seberapa banyak informasi baru yang harus disimpan dalam *unit* LSTM. Gerbang ini berfungsi sebagai *filter* yang memutuskan informasi apa yang harus dimasukkan ke dalam unit LSTM. *Input gate* juga mengambil *input* dari *unit* sebelumnya dan *output* dari *unit* LSTM pada langkah waktu sebelumnya.

Output Gate: Gerbang keluaran mengendalikan seberapa banyak informasi yang akan diungkapkan oleh *unit* LSTM ke *unit* berikutnya dalam urutan data. Gerbang ini menghasilkan *output* aktual dari *unit* LSTM, yang juga bergantung pada *input* dari *unit* sebelumnya dan *output* dari *unit* LSTM pada langkah waktu sebelumnya (Staudemeyer dan Eric, 2019).

Setiap *unit* LSTM juga memiliki sebuah sel memori *internal* yang bertindak sebagai wadah yang menyimpan informasi kontekstual jangka panjang. Sel memori ini dapat menahan informasi penting dari langkah waktu sebelumnya dan memutuskan seberapa banyak informasi yang akan dihapus atau ditambahkan pada langkah waktu saat ini melalui gerbang lupakan dan gerbang masukan.

Keunggulan utama arsitektur LSTM adalah kemampuannya untuk menangani masalah perhitungan yang melibatkan informasi kontekstual jangka panjang. Hal ini membuat LSTM sangat berguna dalam memproses data berurutan yang memiliki dependensi jangka panjang, seperti prediksi sinyal waktu nyata, penerjemahan mesin, dan analisis teks.

2.5 *Preprocessing*

Preprocessing merupakan serangkaian langkah atau teknik yang diterapkan pada data sebelum data tersebut digunakan dalam proses analisis atau pemodelan. Tujuan dari *preprocessing* adalah untuk membersihkan, mentransformasi, dan mempersiapkan data agar sesuai dengan kebutuhan analisis yang akan dilakukan. Beberapa langkah umum dalam *preprocessing* meliputi penghapusan data yang hilang atau tidak *valid*, penanganan *outlier*, normalisasi data, pengkodean variabel kategorikal, dan pemilihan fitur (García dkk., 2015).

Preprocessing memiliki peran penting dalam analisis data karena kualitas data yang buruk atau tidak terstruktur dapat menghasilkan hasil yang tidak akurat atau tidak dapat diandalkan. Dengan melakukan *preprocessing* yang baik, kita dapat meningkatkan kualitas data, mengurangi bias, meningkatkan efisiensi analisis, dan mempersiapkan data untuk diterapkan dalam berbagai algoritma analisis.

Beberapa teknik dan langkah umum dalam *preprocessing* data termasuk:

1. Penghapusan data yang hilang atau tidak *valid*: Data yang hilang atau tidak *valid* dapat mempengaruhi kualitas hasil analisis. Oleh karena itu, langkah-langkah seperti imputasi data hilang atau penghapusan baris atau kolom yang mengandung data yang tidak *valid* dapat diterapkan.
2. Penanganan *outlier*: *Outlier* adalah nilai yang jauh dari pola umum dalam data. *Outlier* dapat mempengaruhi analisis statistik dan pemodelan. Teknik seperti deteksi *outlier* dengan menggunakan metode statistik atau penggunaan teknik penghalusan (*smoothing*) dapat digunakan untuk menangani *outlier*.
3. Normalisasi data: Normalisasi dilakukan untuk mengubah skala data menjadi rentang yang seragam atau untuk menghilangkan perbedaan skala antara fitur-fitur dalam data. Teknik normalisasi yang umum meliputi *min-max scaling*, *z-score normalization*, dan normalisasi *decimal*.
4. Pengkodean variabel kategorikal: Variabel kategorikal perlu dikodekan menjadi bentuk yang dapat dipahami oleh algoritma analisis. Teknik pengkodean seperti *one-hot encoding*, *label encoding*, atau *binary encoding* dapat digunakan untuk mengkonversi variabel kategorikal menjadi bentuk numerik.
5. Pemilihan fitur: Pemilihan fitur adalah proses memilih *subset* fitur yang paling relevan atau informatif dari dataset. Teknik pemilihan fitur seperti metode filter (misalnya, analisis korelasi) atau metode *wrapper* (misalnya, *Recursive Feature Elimination*) dapat digunakan untuk memilih fitur-fitur yang paling penting (García dkk., 2015).

2.6 Bayesian Optimization

Bayesian optimization adalah metode optimasi yang digunakan untuk mencari solusi terbaik dari fungsi objektif yang kompleks dan mahal untuk dievaluasi. Metode ini memadukan pemodelan statistik bayesian dengan teknik pencarian berbasis probabilitas untuk menggantikan metode optimasi tradisional yang mungkin tidak efisien dalam kasus-kasus di mana evaluasi fungsi objektif sangat mahal atau memiliki ruang pencarian yang besar (Shahriari dkk., 2016).

Pendekatan *bayesian optimization* melibatkan dua komponen utama: model *surrogate* dan pemilihan titik evaluasi berdasarkan keputusan yang diambil dari model *surrogate*. Model *surrogate* adalah fungsi aproksimasi yang memodelkan fungsi objektif yang sebenarnya. Model ini dibangun menggunakan teknik seperti *Gaussian Process Regression* (GPR) atau *Random Forests*. Model *surrogate* memberikan perkiraan tentang kinerja fungsi objektif di seluruh ruang pencarian, yang memungkinkan untuk memperoleh informasi tentang di mana mencari solusi yang optimal.

Setelah membangun model *surrogate*, pemilihan titik evaluasi dilakukan berdasarkan keputusan yang diambil dari model tersebut. Biasanya, metode seperti *Upper Confidence Bound* (UCB) atau *Expected Improvement* (EI) digunakan untuk memilih titik evaluasi selanjutnya. Ini memungkinkan untuk memilih titik evaluasi yang memiliki potensi besar untuk menjadi solusi *optimal*, sehingga mempercepat proses optimasi dengan menghindari evaluasi yang tidak perlu.

Salah satu keunggulan utama dari *bayesian optimization* adalah kemampuannya untuk mengatasi masalah dengan ruang pencarian yang kompleks, *non-linear*, dan memiliki banyak *optimum* lokal. *Bayesian optimization* juga dapat digunakan untuk mengoptimalkan *parameter* atau *hyperparameter* dalam algoritma atau model yang sedang diteliti. Pada penelitian ini *Bayesian optimization* digunakan untuk diterapkan pada *hyperparameter tuning* guna mencari *hyperparameter* yang terbaik dengan lebih cepat tanpa melakukan berurutan (Frazier, 2018).

2.7 Drone

Drone adalah sebuah mesin terbang yang berfungsi dengan kendali jarak jauh oleh pilot atau mampu mengendalikan dirinya sendiri yang dioperasikan oleh *operator*, menggunakan hukum aerodinamika untuk mengangkat dirinya, bisa digunakan kembali dan mampu membawa muatan baik senjata maupun muatan lainnya. Penggunaan terbesar dari drone ini adalah di bidang militer, tetapi juga digunakan di bidang geografi, fotografi, dan videografi yang dilakukan secara bebas dan terbuka. Di bidang geografi, drone digunakan sebagai salah satu wahana pengindraan jauh yang sangat penting dalam pembuatan peta, seperti peta penggunaan lahan, peta daerah rawan bencana, dan peta daerah aliran sungai.

Drone ini memiliki 2 model pengendalian, yaitu dengan *program* dan *remote*. Pada drone yang dikendalikan menggunakan *program*, *operator* akan memberikan perintah koordinat beserta aksi yang akan dilakukan drone. Sedangkan drone yang dikendalikan menggunakan *remote*, akan menggunakan frekuensi tertentu untuk melakukan komunikasi antara drone dengan *remote* (Fahlstrom dkk., 2022).

Pada penelitian ini drone digunakan untuk mengambil sampel sinyal yang dipancarkan, lalu disusun menjadi dataset yang akan dilatih pada model yang diusulkan.



SEKOLAH PASCASARJANA